

# **PENGEMBANGAN SISTEM APLIKASI PELAPORAN MASYARAKAT BERBASIS WEB DI KABUPATEN PEKALONGAN**

## **SKRIPSI**

Untuk memenuhi sebagian persyaratan  
memperoleh gelar Sarjana Komputer

Disusun oleh:  
Richa Amalia P  
NIM: 145150200111147



PROGRAM STUDI TEKNIK INFORMATIKA  
JURUSAN TEKNIK INFORMATIKA  
FAKULTAS ILMU KOMPUTER  
UNIVERSITAS BRAWIJAYA  
MALANG  
2018

## PENGESAHAN

PENGEMBANGAN SISTEM APLIKASI PELAPORAN MASYARAKAT BERBASIS WEB DI  
KABUPATEN PEKALONGAN

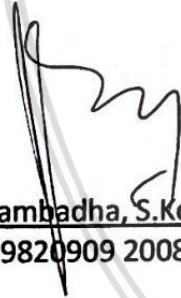
SKRIPSI

Untuk memenuhi sebagian persyaratan  
memperoleh gelar Sarjana Komputer

Disusun Oleh:  
Richa Amalia P  
NIM: 145150200111147

Skripsi ini telah diuji dan dinyatakan lulus pada  
18 Juli 2018  
Telah diperiksa dan disetujui oleh:

Pembimbing I



Bayu Priyambadha, S.Kom., M.Kom.  
NIP. 19820909 200812 1 000

Pembimbing II



Achmad Arwan, S.Kom., M.Kom.  
NIP. 19840815 200812 1 004

Mengetahui

Ketua Jurusan Teknik Informatika



Tri Astoto Kurniawan, S.T., M.T., Ph.D  
NIP. 19710518 200312 1 001





## PERNYATAAN ORISINALITAS

Saya menyatakan dengan sebenar-benarnya bahwa sepanjang pengetahuan saya, di dalam naskah skripsi ini tidak terdapat karya ilmiah yang pernah diajukan oleh orang lain untuk memperoleh gelar akademik di suatu perguruan tinggi, dan tidak terdapat karya atau pendapat yang pernah ditulis atau diterbitkan oleh orang lain, kecuali yang secara tertulis disitasi dalam naskah ini dan disebutkan dalam daftar pustaka.

Apabila ternyata didalam naskah skripsi ini dapat dibuktikan terdapat unsur-unsur plagiasi, saya bersedia skripsi ini digugurkan dan gelar akademik yang telah saya peroleh (sarjana) dibatalkan, serta diproses sesuai dengan peraturan perundang-undangan yang berlaku (UU No. 20 Tahun 2003, Pasal 25 ayat 2 dan Pasal 70).

Malang, 9 Juli 2018



Richa Amalia P

NIM: 145150200111147

## KATA PENGANTAR

Puji syukur kehadiran Allah SWT yang telah melimpahkan rahmat, taufik dan hidayah-Nya sehingga laporan skripsi yang berjudul “Pengembangan Sistem Aplikasi Pelaporan Masyarakat Berbasis Web di Kabupaten Pekalongan” ini dapat terselesaikan.

Penulis menyadari bahwa skripsi ini tidak akan berhasil tanpa bantuan dari beberapa pihak. Oleh karena itu, penulis ingin menyampaikan rasa hormat dan terima kasih kepada:

1. Bapak Bayu Priyambadha, S.Kom., M.Kom. dan Bapak Achmad Arwan, S.Kom., M.Kom. selaku Pembimbing skripsi yang telah dengan sabar membimbing dan mengarahkan penulis sehingga dapat menyelesaikan skripsi ini,
2. Bapak Tri Astoto Kurniawan, S.T, M.T, Ph.D. selaku Ketua Jurusan Teknik Informatika,
3. Bapak Nurul Huda dan Ibu Dwi Sulistyaningsih selaku orang tua penulis, Rasya Amarullah Pramana selaku adik penulis, dan seluruh keluarga besar atas segala nasihat, kasih sayang, perhatian dan kesabarannya di dalam membesarkan dan mendidik penulis, serta senantiasa memberikan doa dan semangat demi terselesainya skripsi ini,
4. Seluruh dosen Teknik Informatika Universitas Brawijaya yang telah banyak memberikan ilmu selama penulis menempuh studi di Teknik Informatika Universitas Brawijaya dan selama penyelesaian skripsi ini,
5. Seluruh teman-teman penulis, khususnya teman-teman kelas Informatika-J, Vivin, Wanda, Farah, Rima, Yati, Elke, yang telah mewarnai hari-hari penulis dan memberikan bantuan morel kepada penulis dalam penyelesaian skripsi ini,
6. Goldy Vito, yang selalu setia menemani penulis dalam suka maupun duka,
7. *Stackoverflow*, yang selalu menjadi penyelamat di kala tak ada satupun yang dapat memberikan solusi dalam penyelesaian skripsi ini.

Penulis menyadari bahwa dalam penyusunan skripsi ini masih banyak kekurangan, sehingga saran dan kritik yang membangun sangat penulis harapkan. Akhir kata penulis berharap skripsi ini dapat membawa manfaat bagi semua pihak yang menggunakannya.

Malang, 9 Juli 2018

Penulis  
Email: richamaliap@student.ub.ac.id

## ABSTRAK

**Richa Amalia P, Pengembangan Sistem Aplikasi Pelaporan Masyarakat Berbasis Web di Kabupaten Pekalongan**

**Pembimbing: Bayu Priyambadha, S.Kom., M.Kom. dan Achmad Arwan, S.Kom.**

Keterbukaan informasi publik merupakan perwujudan demokratisasi dalam pengelolaan pemerintahan yang memungkinkan partisipasi masyarakat dalam mewujudkan *good governance* (tata kelola pemerintahan yang baik). Pada Dinas Komunikasi dan Informatika (Kominfo) Kabupaten Pekalongan, hal itu dilakukan dengan menyediakan wadah yang menampung laporan permasalahan dari masyarakat berupa *form online* yang dapat diakses melalui *website* Pemerintah Kabupaten Pekalongan. Namun, kelemahannya adalah proses distribusi laporan ke Organisasi Perangkat Daerah (OPD) dilakukan secara manual, sehingga waktu penyampaian dan respon cenderung lambat, serta masyarakat tidak mengetahui tindak lanjut dari permasalahan yang disampaikan. Selain itu, tidak ada penilaian terhadap kecepatan masing-masing OPD dalam menjawab laporan yang dapat digunakan sebagai tolak ukur OPD dalam memberikan pelayanan prima kepada masyarakat. Untuk menangani permasalahan tersebut, penulis mengembangkan sistem aplikasi pelaporan masyarakat berbasis web dengan fitur untuk melaporkan masalah di mana masyarakat dapat terhubung dengan OPD yang terkait dengan permasalahannya dan menampilkan penilaian kinerja tiap OPD berdasarkan kecepatan responnya terhadap laporan yang disampaikan masyarakat. Penulis menggunakan *waterfall model* dalam mengembangkan perangkat lunak. Berdasarkan pengujian yang dilakukan, pengujian unit, pengujian integrasi, dan pengujian validasi menghasilkan nilai 100% valid pada semua kasus uji. Sedangkan pengujian *compatibility* menunjukkan bahwa sistem dapat berjalan dengan baik pada 86,67% dari 13 versi *browser* yang diujikan.

Kata kunci: keterbukaan informasi publik, sistem pelaporan, penilaian kinerja, rekayasa perangkat lunak, *waterfall model*

## ABSTRACT

**Richa Amalia P, Pengembangan Sistem Aplikasi Pelaporan Masyarakat Berbasis Web di Kabupaten Pekalongan**

**Adviser: Bayu Priyambadha, S.Kom., M.Kom. dan Achmad Arwan, S.Kom.**

*Public information disclosure is a manifestation of democratization in the management of governance that allows public participation in realizing good governance. In the Office of Communications and Informatics (Kominfo) of Pekalongan Regency, it is done by providing a media that accommodates reports of problems from the citizens in an online form which can be accessed through the website of the Government of Pekalongan Regency. However, the problem is the process of distributing reports to Local Government Agency (OPD) is done manually, so that the delivery and response time tend to be slow, and the citizens do not know the follow-up of the problem conveyed. In addition, there is no assessment of the speed of OPD in answering reports that can be used as a benchmark of the OPD in providing excellent service to the citizens. To solve the problem, the author developed a web-based reporting system to report a problem where people can connect with the OPD correlated with the problem and display performance assessments for each OPD based on the speed of their response to reports submitted by the citizens. The author used waterfall model in developing the software. Based on the testing performed, the unit testing, integration testing, and validation testing result 100% valid in all test cases. And the compatibility testing shows that the system can run well on 86.67% of 13 browser versions.*

**Keywords:** *public information disclosure, reporting system, performance assessment, software engineering, waterfall model*

## DAFTAR ISI

PENGESAHAN .....	ii
PERNYATAAN ORISINALITAS .....	iii
KATA PENGANTAR.....	iv
ABSTRAK.....	v
ABSTRACT .....	vi
DAFTAR ISI .....	vii
DAFTAR TABEL.....	xi
DAFTAR GAMBAR .....	xv
DAFTAR LAMPIRAN .....	xvii
BAB 1 PENDAHULUAN.....	1
1.1 Latar Belakang.....	1
1.2 Rumusan Masalah.....	2
1.3 Tujuan .....	3
1.4 Manfaat.....	3
1.5 Batasan Masalah.....	4
1.6 Sistematika Pembahasan.....	4
BAB 2 LANDASAN KEPUSTAKAAN .....	6
2.1 Kajian Pustaka .....	6
2.2 Pelaporan Masyarakat.....	6
2.2.1 Penanganan Pelaporan Masyarakat .....	7
2.3 Penilaian Kinerja .....	7
2.3.1 Konsep Penilaian Kinerja.....	7
2.3.2 Metode Penilaian .....	8
2.4 <i>Software Development Life Cycle (SDLC)</i> .....	8
2.4.1 <i>Waterfall Model</i> .....	8
2.5 Pendekatan Berorientasi Objek .....	10
2.5.1 Pemodelan .....	10
2.5.2 UML .....	10
2.6 Pengujian .....	15
2.7 Teknologi Pengembangan Sistem.....	16



2.7.1 CodeIgniter Framework .....	16
2.7.2 SortSite Tool .....	17
BAB 3 METODOLOGI PENELITIAN .....	18
3.1 Studi Kepustakaan .....	18
3.2 Analisis Kebutuhan .....	19
3.3 Perancangan .....	19
3.4 Implementasi .....	20
3.5 Pengujian .....	20
3.6 Kesimpulan dan Saran .....	20
BAB 4 ANALISIS KEBUTUHAN .....	21
4.1 Penanganan Pelaporan Masyarakat di Kabupaten Pekalongan .....	21
4.2 Gambaran Umum Sistem .....	22
4.3 Identifikasi Aktor .....	23
4.4 Identifikasi Kebutuhan .....	24
4.4.1 Kebutuhan Fungsional .....	24
4.4.2 Kebutuhan Non-Fungsional .....	30
4.5 Analisis Data .....	30
4.6 Use Case Diagram .....	31
4.7 Use Case Scenario .....	32
4.7.1 Mendaftar .....	32
4.7.2 Login .....	34
4.7.3 Mencari Laporan .....	35
4.7.4 Melihat Status Tahapan Laporan .....	35
4.7.5 Melihat Jawaban Laporan .....	36
4.7.6 Melihat Statistik Kinerja OPD .....	36
4.7.7 Mengubah Bulan dan Tahun Statistik .....	37
4.7.8 Melihat Rekapitulasi Laporan .....	37
4.7.9 Mengubah Bulan dan Tahun Rekapitulasi .....	38
4.7.10 Melaporkan Permasalahan .....	38
4.7.11 Melihat Riwayat Laporan .....	39
4.7.12 Memberi Respon Balik .....	40
4.7.13 Menutup Laporan .....	40

4.7.14 Menjawab Laporan .....	41
4.7.15 Melihat OPD .....	42
4.7.16 Menambah OPD .....	42
4.7.17 Menghapus OPD .....	43
4.7.18 Mengubah OPD .....	43
4.7.19 Melakukan Moderasi Laporan .....	44
4.7.20 Meneruskan Laporan .....	44
4.7.21 Logout .....	45
<b>BAB 5 PERANCANGAN DAN IMPLEMENTASI .....</b>	<b>46</b>
5.1 Perancangan .....	46
5.1.1 Perancangan Data .....	46
5.1.2 Pemodelan <i>Sequence Diagram</i> .....	47
5.1.3 Pemodelan <i>Class Diagram</i> .....	50
5.1.4 Perancangan Algoritme.....	50
5.1.5 Perancangan Antarmuka.....	52
5.2 Implementasi .....	73
5.2.1 Spesifikasi Pengembangan Sistem .....	73
5.2.2 Implementasi Data .....	74
5.2.3 Implementasi Kode Program .....	75
5.2.4 Implementasi Antarmuka .....	79
<b>BAB 6 PENGUJIAN .....</b>	<b>89</b>
6.1 Pengujian Unit.....	89
6.1.1 Pengujian Unit <i>Method</i> hitungStatistik.....	89
6.1.2 Pengujian Unit <i>Method</i> hitungRekap.....	92
6.1.3 Pengujian Unit <i>Method</i> teruskan .....	95
6.2 Pengujian Integrasi .....	96
6.3 Pengujian Validasi .....	101
6.3.1 Pengujian Validasi Mendaftar .....	101
6.3.2 Pengujian Validasi <i>Login</i> .....	106
6.3.3 Pengujian Validasi Mencari Laporan .....	108
6.3.4 Pengujian Validasi Melihat Status Tahapan Laporan .....	108
6.3.5 Pengujian Validasi Melihat Jawaban Laporan.....	109

6.3.6 Pengujian Validasi Melihat Statistik Kinerja OPD .....	109
6.3.7 Pengujian Validasi Mengubah Bulan dan Tahun Statistik.....	109
6.3.8 Pengujian Validasi Melihat Rekapitulasi Laporan .....	110
6.3.9 Pengujian Validasi Mengubah Bulan dan Tahun Rekapitulasi ..	111
6.3.10 Pengujian Validasi Melaporkan Permasalahan.....	111
6.3.11 Pengujian Validasi Melihat Riwayat Laporan .....	113
6.3.12 Pengujian Validasi Memberi Respon Balik.....	113
6.3.13 Pengujian Validasi Menutup Laporan .....	114
6.3.14 Pengujian Validasi Menjawab Laporan .....	115
6.3.15 Pengujian Validasi Melihat OPD.....	117
6.3.16 Pengujian Validasi Menambah OPD.....	117
6.3.17 Pengujian Validasi Menghapus OPD .....	118
6.3.18 Pengujian Validasi Mengubah OPD.....	119
6.3.19 Pengujian Validasi Melakukan Moderasi Laporan .....	120
6.3.20 Pengujian Validasi Meneruskan Laporan .....	120
6.3.21 Pengujian Validasi <i>Logout</i> .....	121
6.4 Pengujian <i>Compatibility</i> .....	122
BAB 7 PENUTUP .....	123
7.1 Kesimpulan.....	123
7.2 Saran .....	124
DAFTAR PUSTAKA.....	125
LAMPIRAN A HASIL WAWANCARA.....	127

## DAFTAR TABEL

Tabel 2.1 Simbol Pada Class Diagram .....	12
Tabel 2.2 Simbol Pada Use Case Diagram .....	13
Tabel 2.3 Simbol Pada Sequence Diagram .....	14
Tabel 4.1 Identifikasi Aktor .....	23
Tabel 4.2 Kebutuhan Fungsional .....	24
Tabel 4.3 Spesifikasi Kebutuhan Fungsional .....	26
Tabel 4.4 Kebutuhan Non-Fungsional .....	30
Tabel 4.5 <i>Use Case Scenario</i> Mendaftar .....	32
Tabel 4.6 <i>Use Case Scenario</i> Login .....	34
Tabel 4.7 <i>Use Case Scenario</i> Mencari Laporan .....	35
Tabel 4.8 <i>Use Case Scenario</i> Melihat Status Tahapan Laporan .....	35
Tabel 4.9 <i>Use Case Scenario</i> Melihat Jawaban Laporan .....	36
Tabel 4.10 <i>Use Case Scenario</i> Melihat Statistik Kinerja OPD .....	36
Tabel 4.11 <i>Use Case Scenario</i> Mengubah Bulan dan Tahun Statistik .....	37
Tabel 4.12 <i>Use Case Scenario</i> Melihat Rekapitulasi Laporan .....	37
Tabel 4.13 <i>Use Case Scenario</i> Melaporkan Permasalahan .....	38
Tabel 4.14 <i>Use Case Scenario</i> Melihat Riwayat Laporan .....	39
Tabel 4.15 <i>Use Case Scenario</i> Memberi Respon Balik .....	40
Tabel 4.16 <i>Use Case Scenario</i> Menutup Laporan .....	40
Tabel 4.17 <i>Use Case Scenario</i> Menjawab Laporan .....	41
Tabel 4.18 <i>Use Case Scenario</i> Melihat OPD .....	42
Tabel 4.19 <i>Use Case Scenario</i> Menambah OPD .....	42
Tabel 4.20 <i>Use Case Scenario</i> Menghapus OPD .....	43
Tabel 4.21 <i>Use Case Scenario</i> Mengubah OPD .....	43
Tabel 4.22 <i>Use Case Scenario</i> Melakukan Moderasi Laporan .....	44
Tabel 4.23 <i>Use Case Scenario</i> Logout .....	45
Tabel 5.1 <i>Pseudocode</i> Melihat Statistik .....	50
Tabel 5.2 <i>Pseudocode</i> Melihat Rekapitulasi Laporan .....	51
Tabel 5.3 <i>Pseudocode</i> Meneruskan Laporan .....	52
Tabel 5.4 Penjelasan Perancangan Antarmuka Halaman Beranda <i>User</i> .....	53

Tabel 5.5 Penjelasan Perancangan Antarmuka Halaman Daftar .....	54
Tabel 5.6 Penjelasan Perancangan Antarmuka Halaman Cari Laporan.....	55
Tabel 5.7 Penjelasan Perancangan Antarmuka Halaman Statistik .....	56
Tabel 5.8 Penjelasan Perancangan Antarmuka Halaman Rekapitulasi Laporan ..	58
Tabel 5.9 Penjelasan Perancangan Antarmuka Halaman Formulir Pelaporan .....	59
Tabel 5.10 Penjelasan Perancangan Antarmuka Halaman Riwayat Laporan .....	60
Tabel 5.11 Penjelasan Perancangan Antarmuka Detail Laporan <i>Member</i> .....	61
Tabel 5.12 Penjelasan Perancangan Antarmuka Halaman Beranda Admin .....	63
Tabel 5.13 Penjelasan Perancangan Antarmuka Halaman Moderasi Laporan.....	64
Tabel 5.14 Penjelasan Perancangan Antarmuka Meneruskan Laporan .....	65
Tabel 5.15 Penjelasan Perancangan Antarmuka Halaman Kelola Daftar OPD .....	66
Tabel 5.16 Penjelasan Perancangan Antarmuka Halaman Edit OPD .....	68
Tabel 5.17 Penjelasan Perancangan Antarmuka Halaman Tambah OPD .....	69
Tabel 5.18 Penjelasan Perancangan Antarmuka Halaman Beranda OPD .....	70
Tabel 5.19 Penjelasan Perancangan Antarmuka Halaman Laporan Masuk OPD .	71
Tabel 5.20 Penjelasan Perancangan Antarmuka Halaman Rincian Laporan OPD	72
Tabel 5.21 Spesifikasi Perangkat Keras .....	74
Tabel 5.22 Spesifikasi Perangkat Lunak .....	74
Tabel 5.23 Implementasi Data .....	74
Tabel 5.24 <i>Source Code</i> Melihat Statistik.....	75
Tabel 5.25 <i>Source Code</i> Melihat Rekapitulasi Laporan .....	77
Tabel 5.26 <i>Source Code</i> Meneruskan Laporan.....	78
Tabel 6.1 <i>Pseudocode</i> Pengujian <i>Method</i> hitungStatistik.....	89
Tabel 6.2 Hasil Pengujian Unit <i>Method</i> hitungStatistik .....	91
Tabel 6.3 <i>Pseudocode Method</i> hitungRekap .....	92
Tabel 6.4 Hasil Pengujian Unit <i>Method</i> hitungRekap .....	94
Tabel 6.5 <i>Pseudocode Method</i> teruskan.....	95
Tabel 6.6 Hasil Pengujian Unit <i>Method</i> teruskan.....	96
Tabel 6.7 Langkah Uji Pengujian Integrasi .....	97
Tabel 6.8 <i>Source Code</i> Pengujian <i>Method</i> stubInsertData dengan Nilai opd “1” .	97
Tabel 6.9 <i>Source Code</i> Pengujian <i>Method</i> stubInsertData dengan Nilai opd “0” .	98
Tabel 6.10 <i>Source Code Method</i> teruskan .....	99



Tabel 6.11 Hasil Pengujian Integrasi <i>Method</i> insertData.....	100
Tabel 6.12 Kasus Uji Mendaftar .....	101
Tabel 6.13 Kasus Uji Mendaftar Alternatif 1 .....	101
Tabel 6.14 Kasus Uji Mendaftar Alternatif 2 .....	102
Tabel 6.15 Kasus Uji Mendaftar Alternatif 3 .....	102
Tabel 6.16 Kasus Uji Mendaftar Alternatif 4 .....	103
Tabel 6.17 Kasus Uji Mendaftar Alternatif 5 .....	103
Tabel 6.18 Kasus Uji Mendaftar Alternatif 6 .....	104
Tabel 6.19 Kasus Uji Mendaftar Alternatif 7 .....	104
Tabel 6.20 Kasus Uji Mendaftar Alternatif 8 .....	105
Tabel 6.21 Kasus Uji Mendaftar Alternatif 9 .....	105
Tabel 6.22 Kasus Uji <i>Login</i> Sebagai <i>Member</i> .....	106
Tabel 6.23 Kasus Uji <i>Login</i> Sebagai Admin .....	106
Tabel 6.24 Kasus Uji <i>Login</i> Sebagai OPD .....	106
Tabel 6.25 Kasus Uji <i>Login</i> Alternatif 1 .....	107
Tabel 6.26 Kasus Uji <i>Login</i> Alternatif 2 .....	107
Tabel 6.27 Kasus Uji <i>Login</i> Alternatif 3 .....	107
Tabel 6.28 Kasus Uji Mencari Laporan .....	108
Tabel 6.29 Kasus Uji Mencari Laporan Alternatif 1 .....	108
Tabel 6.30 Kasus Uji Melihat Status Tahapan Laporan .....	108
Tabel 6.31 Kasus Uji Melihat Jawaban Laporan .....	109
Tabel 6.32 Kasus Uji Melihat Statistik Kinerja OPD .....	109
Tabel 6.33 Kasus Uji Mengubah Bulan dan Tahun Statistik .....	109
Tabel 6.34 Kasus Uji Mengubah Bulan dan Tahun Statistik Alternatif 1 .....	110
Tabel 6.35 Kasus Uji Melihat Rekapitulasi Laporan .....	110
Tabel 6.36 Kasus Uji Mengubah Bulan dan Tahun Rekapitulasi .....	111
Tabel 6.37 Kasus Uji Mengubah Bulan dan Tahun Rekapitulasi Alternatif 1 .....	111
Tabel 6.38 Kasus Uji Melaporkan Permasalahan .....	111
Tabel 6.39 Kasus Uji Melaporkan Permasalahan Alternatif 1 .....	112
Tabel 6.40 Kasus Uji Melaporkan Permasalahan Alternatif 2 .....	112
Tabel 6.41 Kasus Uji Melihat Riwayat Laporan .....	113
Tabel 6.42 Kasus Uji Memberi Respon Balik .....	113

Tabel 6.43 Kasus Uji Memberi Respon Balik Alternatif 1.....	114
Tabel 6.44 Kasus Uji Memberi Respon Balik Alternatif 2.....	114
Tabel 6.45 Kasus Uji Menutup Laporan .....	114
Tabel 6.46 Kasus Uji Menutup Laporan Alternatif 1.....	115
Tabel 6.47 Kasus Uji Menjawab Laporan .....	115
Tabel 6.48 Kasus Uji Menjawab Laporan Alternatif 1.....	116
Tabel 6.49 Kasus Uji Menjawab Laporan Alternatif 2.....	116
Tabel 6.50 Kasus Uji Melihat OPD .....	117
Tabel 6.51 Kasus Uji Menambah OPD.....	117
Tabel 6.52 Kasus Uji Menambah OPD Alternatif 1 .....	118
Tabel 6.53 Kasus Uji Menghapus OPD .....	118
Tabel 6.54 Kasus Uji Menghapus OPD Alternatif 1.....	119
Tabel 6.55 Kasus Uji Mengubah OPD.....	119
Tabel 6.56 Kasus Uji Melakukan Moderasi Laporan .....	120
Tabel 6.57 Kasus Uji Melakukan Moderasi Laporan Alternatif 1.....	120
Tabel 6.58 Kasus Uji Meneruskan Laporan .....	121
Tabel 6.59 Kasus Uji Meneruskan Laporan Alternatif 1.....	121
Tabel 6.60 Kasus Uji <i>Logout</i> .....	121

## DAFTAR GAMBAR

Gambar 2.1 Ilustrasi <i>Waterfall Model</i> .....	9
Gambar 2.2 Diagram UML.....	11
Gambar 3.1 Diagram Alir Metode Penelitian .....	18
Gambar 4.1 Diagram Aktivitas Pelaporan Masyarakat Saat Ini .....	21
Gambar 4.2 Diagram Aktivitas Pelaporan Masyarakat Yang Ditawarkan.....	23
Gambar 4.3 <i>Entity Relationship Diagram</i> (ERD).....	31
Gambar 4.4 <i>Use Case Diagram</i> .....	32
Gambar 5.1 <i>Physical Data Model</i> (PDM) .....	46
Gambar 5.2 <i>Sequence Diagram</i> Melihat Statistik Kinerja OPD.....	47
Gambar 5.3 <i>Sequence Diagram</i> Melihat Rekapitulasi Laporan .....	48
Gambar 5.4 <i>Sequence Diagram</i> Meneruskan Laporan .....	49
Gambar 5.5 <i>Class Diagram</i> .....	50
Gambar 5.6 Perancangan Antarmuka Halaman Beranda <i>User</i> .....	53
Gambar 5.7 Perancangan Antarmuka Halaman Daftar .....	54
Gambar 5.8 Perancangan Antarmuka Halaman Cari Laporan .....	55
Gambar 5.9 Perancangan Antarmuka Halaman Statistik .....	56
Gambar 5.10 Perancangan Antarmuka Halaman Rekapitulasi Laporan.....	57
Gambar 5.11 Perancangan Antarmuka Halaman Formulir Pelaporan .....	59
Gambar 5.12 Perancangan Antarmuka Halaman Riwayat Laporan .....	60
Gambar 5.13 Perancangan Antarmuka Detail Laporan <i>Member</i> .....	61
Gambar 5.14 Perancangan Antarmuka Halaman Beranda Admin .....	63
Gambar 5.15 Perancangan Antarmuka Halaman Moderasi Laporan .....	64
Gambar 5.16 Perancangan Antarmuka Meneruskan Laporan .....	65
Gambar 5.17 Perancangan Antarmuka Halaman Kelola Daftar OPD .....	66
Gambar 5.18 Perancangan Antarmuka Halaman Edit OPD .....	68
Gambar 5.19 Perancangan Antarmuka Halaman Tambah OPD .....	69
Gambar 5.20 Perancangan Antarmuka Halaman Beranda OPD.....	70
Gambar 5.21 Perancangan Antarmuka Halaman Laporan Masuk OPD .....	71
Gambar 5.22 Perancangan Antarmuka Halaman Rincian Laporan OPD .....	72
Gambar 5.23 Implementasi Antarmuka Halaman Beranda <i>User</i> .....	80

Gambar 5.24 Implementasi Antarmuka Halaman Daftar .....	80
Gambar 5.25 Implementasi Antarmuka Halaman Cari Laporan .....	81
Gambar 5.26 Implementasi Antarmuka Halaman Statistik .....	81
Gambar 5.27 Implementasi Antarmuka Halaman Rekapitulasi Laporan .....	82
Gambar 5.28 Implementasi Antarmuka Halaman Formulir Pelaporan .....	82
Gambar 5.29 Implementasi Antarmuka Halaman Riwayat Laporan .....	83
Gambar 5.30 Implementasi Antarmuka Halaman Rincian Laporan <i>Member</i> .....	83
Gambar 5.31 Implementasi Antarmuka Halaman Beranda Admin .....	84
Gambar 5.32 Implementasi Antarmuka Halaman Moderasi Laporan .....	84
Gambar 5.33 Implementasi Antarmuka Halaman Meneruskan Laporan .....	85
Gambar 5.34 Implementasi Antarmuka Halaman Kelola Daftar OPD .....	85
Gambar 5.35 Implementasi Antarmuka Halaman Edit OPD .....	86
Gambar 5.36 Implementasi Antarmuka Halaman Tambah OPD .....	86
Gambar 5.37 Implementasi Antarmuka Halaman Beranda OPD .....	87
Gambar 5.38 Implementasi Antarmuka Halaman Laporan Masuk .....	87
Gambar 5.39 Implementasi Antarmuka Halaman Laporan Masuk .....	88
Gambar 6.1 <i>Flow Graph Method</i> hitungStatistik .....	90
Gambar 6.2 <i>Flow Graph Method</i> hitungRekap .....	93
Gambar 6.3 <i>Flow Graph Method</i> teruskan .....	95
Gambar 6.4 Diagram Hierarki Pengujian Integrasi .....	97
Gambar 6.5 Hasil Pengujian <i>Method</i> stubInsertData dengan Nilai opd "1" .....	98
Gambar 6.6 Hasil Pengujian <i>Method</i> stubInsertData dengan Nilai opd "0" .....	98
Gambar 6.7 <i>Flow Graph</i> Integrasi <i>Method</i> insertData .....	99
Gambar 6.8 Hasil Pengujian Integrasi <i>Method</i> insertData dengan Nilai "1" .....	100
Gambar 6.9 Hasil Pengujian Integrasi <i>Method</i> insertData dengan Nilai "0" .....	101
Gambar 6.10 Hasil Pengujian <i>Compatibility</i> .....	122

## DAFTAR LAMPIRAN

LAMPIRAN A HASIL WAWANCARA.....	127
---------------------------------	-----





## BAB 1 PENDAHULUAN

### 1.1 Latar Belakang

Keterbukaan informasi publik merupakan aspek penting dalam pemenuhan hak individu atas informasi publik sebagaimana diatur di dalam Undang-Undang Dasar Tahun 1945 Pasal 28F. Masyarakat memiliki hak untuk memperoleh informasi publik dan pemerintah sebagai badan publik memiliki kewajiban untuk membangun dan mengembangkan sistem informasi untuk mengelola informasi publik sehingga dapat diakses dengan mudah. Hal ini telah diatur dalam Undang-Undang No. 14 Tahun 2008 Tentang Keterbukaan Informasi Publik pada Bab (III) tentang hak dan kewajiban, baik hak dan kewajiban yang dimiliki masyarakat sebagai pemohon informasi maupun hak dan kewajiban yang dimiliki badan publik sebagai penyedia informasi. Keterbukaan informasi publik merupakan perwujudan demokratisasi dalam pengelolaan pemerintahan yang memungkinkan partisipasi masyarakat dalam mewujudkan *good governance*, yaitu tata kelola pemerintahan yang baik (Informasi, 2013). Prinsip-prinsip pemerintahan yang baik sebagaimana tertulis pada Peraturan Pemerintah No. 101 Tahun 2000 meliputi:

1. Profesionalitas;
2. Akuntabilitas;
3. Transparansi;
4. Pelayanan prima;
5. Demokrasi dan partisipasi;
6. Efisiensi dan efektivitas;
7. Supremasi hukum.

Keterbukaan informasi publik mengandung beberapa prinsip yang dapat mewujudkan *good governance*, yaitu akuntabilitas, transparansi, serta demokrasi dan partisipasi.

Perkembangan teknologi informasi yang semakin pesat kini telah memberikan manfaat ke hampir seluruh sektor kehidupan manusia, tak terkecuali sektor pemerintahan. Teknologi informasi di lingkungan pemerintahan dimanfaatkan sebagai alat bantu berbagai urusan pemerintahan, mulai dari administrasi, pengarsipan, hingga pelayanan publik (Idris, 2015). Begitu pula pada Dinas Komunikasi dan Informatika (Kominfo) Kabupaten Pekalongan yang memanfaatkan teknologi informasi dalam urusan pemerintahannya, salah satunya yaitu dengan menyediakan wadah yang menampung laporan permasalahan dari masyarakat. Hal ini merupakan salah satu perwujudan dari Dinas Kominfo Kabupaten Pekalongan dalam pemenuhan kewajibannya terhadap Undang-Undang Keterbukaan Informasi Publik (KIP) seperti yang telah penulis jabarkan di atas. Wadah tersebut berupa *form online* yang dapat diakses masyarakat melalui halaman "Suara Warga" pada *website* Pemerintah Kabupaten Pekalongan

([www.pekalongankab.go.id](http://www.pekalongankab.go.id)). Namun, sistem tersebut masih memiliki kelemahan, yaitu tidak adanya integrasi antara sistem dengan Organisasi Perangkat Daerah (OPD) yang terkait dengan permasalahan yang diadukan masyarakat. Berdasarkan Peraturan Pemerintah No. 8 Tahun 2003, OPD merupakan organisasi/lembaga pada pemerintah daerah yang bertanggungjawab kepada Kepala Daerah dalam rangka penyelenggaraan pemerintahan. Dalam sistem pelaporan di Pemerintah Kabupaten Pekalongan saat ini, laporan permasalahan dari masyarakat didistribusikan ke OPD terkait secara kolektif dan manual, sehingga proses penyampaian cenderung lama. Penyampaian laporan yang lama berdampak pada respon dari pemerintah ke masyarakat menjadi lambat. Selain itu, proses distribusi laporan yang dilakukan secara manual menyebabkan masyarakat tidak mengetahui bagaimana tindak lanjut pemerintah terhadap laporan tersebut. Hal ini tentunya menghambat tujuan Dinas Kominfo Kabupaten Pekalongan dalam mewujudkan Undang-Undang Keterbukaan Informasi Publik. Permasalahan juga terjadi karena tidak adanya penilaian terhadap kecepatan masing-masing OPD dalam menjawab laporan yang dapat digunakan sebagai tolak ukur OPD dalam memberikan pelayanan prima kepada masyarakat (Arif, et al., 2017).

Berangkat dari permasalahan yang telah dijabarkan, penulis melakukan pengembangan sebuah sistem aplikasi sebagai “wadah” baru yang dapat memungkinkan masyarakat untuk memberikan pesan berupa laporan masalah, keluhan, kritik, atau saran yang ingin disampaikan kepada Pemerintah Kabupaten Pekalongan. Fitur ini dapat menghubungkan masyarakat dengan OPD yang terkait dengan permasalahan yang disampaikan oleh masyarakat, sehingga OPD dapat langsung menerima dan memberikan respon terhadap permasalahan tersebut. Masyarakat dapat melihat respon yang diberikan OPD dan progres dari pemecahan masalah yang perlu ditindaklanjuti. Aplikasi ini juga dapat memberikan penilaian kinerja tiap OPD berdasarkan kecepatan responnya terhadap laporan yang disampaikan masyarakat, di mana masyarakat dapat melihat penilaian tersebut. Hal ini ditujukan untuk memberikan motivasi kepada tiap OPD untuk berlomba memberikan pelayanan prima kepada masyarakat dan sebagai upaya peningkatan kinerja tiap OPD. Di dalam aplikasi ini, pengguna dibagi menjadi 4 aktor, yaitu *user*, *member*, *admin*, dan OPD. Aplikasi yang dikembangkan merupakan aplikasi berbasis web. Pemilihan *platform* web didasarkan pada kemudahan pengguna untuk mengakses aplikasi ini kapan saja dan di mana saja. Selain itu, pilihan gawai yang dapat digunakan beragam, pengguna dapat menggunakan komputer maupun *handphone* untuk mengakses aplikasi ini dengan melalui *web browser*. Selanjutnya, aplikasi ini akan dikembangkan dalam penelitian skripsi yang penulis beri judul “Pengembangan Sistem Aplikasi Pelaporan Masyarakat Berbasis Web di Kabupaten Pekalongan”.

## 1.2 Rumusan Masalah

Berdasarkan penjabaran dari latar belakang, maka dapat dirumuskan permasalahan sebagai berikut:

1. Bagaimana hasil analisis kebutuhan sistem aplikasi pelaporan masyarakat berbasis web di Kabupaten Pekalongan?
2. Bagaimana hasil perancangan sistem aplikasi pelaporan masyarakat berbasis web di Kabupaten Pekalongan?
3. Bagaimana hasil implementasi sistem aplikasi pelaporan masyarakat berbasis web di Kabupaten Pekalongan?
4. Bagaimana hasil pengujian sistem aplikasi pelaporan masyarakat berbasis web di Kabupaten Pekalongan?

### 1.3 Tujuan

Adapun tujuan diadakannya penelitian ini adalah:

1. Menganalisis kebutuhan sistem aplikasi pelaporan masyarakat berbasis web di Kabupaten Pekalongan.
2. Merancang sistem aplikasi pelaporan masyarakat berbasis web di Kabupaten Pekalongan.
3. Mengimplementasikan sistem aplikasi pelaporan masyarakat berbasis web di Kabupaten Pekalongan.
4. Menguji sistem aplikasi pelaporan masyarakat berbasis web di Kabupaten Pekalongan.

### 1.4 Manfaat

Manfaat yang didapat dari pengembangan aplikasi ini adalah sebagai berikut:

1. Bagi Dinas Kominfo

Sistem ini diharapkan dapat mendukung Dinas Kominfo dalam memenuhi kewajibannya untuk menyediakan sarana bagi masyarakat dalam melaporkan permasalahan yang berkaitan dengan publik.

2. Bagi Pemerintah Kabupaten Pekalongan

Sistem ini diharapkan dapat mendukung Pemerintah Kabupaten Pekalongan dalam menyelenggarakan pemerintahan yang baik dengan adanya keterbukaan informasi publik.

3. Bagi masyarakat

Sistem ini diharapkan dapat memenuhi kebutuhan masyarakat terhadap keterbukaan informasi publik, di mana masyarakat dapat melaporkan permasalahan yang ditemui dan mengetahui respon dari pelaporan tersebut.

4. Bagi penulis

Proses pengembangan sistem ini diharapkan dapat menambah wawasan keilmuan dan pengalaman penelitian penulis, serta dapat

mengimplementasikan ilmu yang telah didapat di dunia nyata yang diharapkan dapat berguna bagi masyarakat.

## 1.5 Batasan Masalah

Untuk memperoleh hasil pembahasan yang sesuai dengan yang diharapkan, maka perlu dilakukan pembatasan masalah terhadap pengembangan aplikasi ini, di antaranya:

1. Objek penelitian dan sumber data yang digunakan hanya berasal dari Kabupaten Pekalongan.
2. Pembahasan difokuskan pada rekayasa perangkat lunak dari pengembangan sistem aplikasi pelaporan masyarakat berbasis web di Kabupaten Pekalongan.
3. Sistem berjalan setiap hari dan tidak dapat mendeteksi hari libur nasional.
4. Aplikasi yang akan dikembangkan berupa *web based application*.
5. Pengembangan aplikasi ini menggunakan bahasa pemrograman HTML, CSS, PHP, dan JavaScript.
6. Framework yang digunakan untuk mengembangkan aplikasi ini yaitu *CodeIgniter 3.1.6*.
7. Pengembangan aplikasi ini menggunakan web server *Apache*.
8. *Code editor* yang digunakan dalam proses pengembangan aplikasi ini adalah Notepad++.
9. *Database* yang digunakan dalam mengembangkan aplikasi ini adalah MySQL.
10. SDLC (*Software Development Life Cycle*) yang digunakan pada pengembangan aplikasi ini adalah *Waterfall Model*.

## 1.6 Sistematika Pembahasan

Sistematika pembahasan ditujukan untuk memberikan gambaran dari penyusunan skripsi ini secara umum dan garis besar yang terdiri dari beberapa bab, sebagai berikut:

### Bab I Pendahuluan

Bab Pendahuluan berisi penjelasan mengenai latar belakang masalah, rumusan masalah, tujuan penelitian, manfaat penelitian, batasan masalah, dan sistematika pembahasan.

### Bab II Landasan Kepustakaan

Landasan Kepustakaan berisi uraian dan pembahasan tentang teori, konsep, model, metode, atau sistem dari literatur ilmiah. Dalam landasan kepustakaan terdapat landasan teori dan referensi yang mendasari pengembangan sistem aplikasi pelaporan masyarakat ini.

### Bab III Metodologi Penelitian

Berisi penjelasan metode dan langkah kerja yang dilakukan dalam proses pengembangan aplikasi.

#### **Bab IV Analisis Kebutuhan**

Membahas analisis kebutuhan dari aplikasi sesuai dengan dasar teori dan literatur yang ada.

#### **Bab V Perancangan dan Implementasi**

Membahas perancangan aplikasi dan implementasi dari sistem aplikasi pelaporan masyarakat sesuai dengan perancangan yang telah disusun.

#### **Bab VI Pengujian**

Memuat hasil pengujian dan analisis terhadap aplikasi yang telah dikembangkan.

#### **Bab VII Penutup**

Memuat kesimpulan yang diperoleh dari pengembangan aplikasi dalam penelitian skripsi serta saran untuk pengembangan lebih lanjut.





## BAB 2 LANDASAN KEPUSTAKAAN

Berdasarkan judul penelitian yang disusun oleh penulis yaitu “Pengembangan Sistem Aplikasi Pelaporan Masyarakat Berbasis Web di Kabupaten Pekalongan”, maka dibutuhkan penjelasan mengenai literatur yang digunakan pada penelitian ini.

### 2.1 Kajian Pustaka

Dalam melakukan penelitian ini, penulis mengacu pada penelitian terkait yang pernah dilakukan sebelumnya. Penelitian pertama berjudul “*Web Application for Complaint Tracking and Resolving*” yang ditulis oleh Sneha Alve, Vishakha Babardesai, Sneha Bhosale, Siddhi Kapadi, dan Prof. Atul. B. Yadav dalam International Research Journal of Engineering and Technology (IRJET) Volume: 04 Issue: 04 pada April 2017. Penelitian tersebut membahas tentang pengembangan aplikasi web penyelesaian dan pelacakan komplain yang ditujukan untuk membuat komplain lebih mudah untuk dikoordinir, dipantau, dilacak, dan diselesaikan. Hasil yang diperoleh dari penelitian tersebut adalah aplikasi web yang dibangun dapat mengatasi permasalahan yang dihadapi dalam menangani keluhan dengan tepat waktu. Penelitian kedua yaitu “Pembuatan Website Monitoring Keluhan Pelanggan Pada PT. Telkom Unit Layanan Ngadirojo” yang ditulis oleh Anggi Bingar Kusuma dan Lies Yulianto pada *Indonesian Journal on Networking and Security (IJNS)* Volume 3 No. 4 Tahun 2014. Penelitian tersebut bertujuan untuk mempermudah proses penyaluran informasi antara pelanggan dengan pihak PT. Telkom Unit Layanan Ngadirojo dengan pembuatan website *monitoring* keluhan pelanggan. Hasil yang diperoleh dari penelitian tersebut adalah pihak PT. Telkom Unit Layanan Ngadirojo mendapatkan kemudahan dalam memantau keluhan-keluhan dan melakukan *update* informasi melalui *website*, serta pelanggan dapat melaporkan keluhannya secara *online* yang mana data tersebut langsung tersimpan untuk mempermudah pada saat pencarian keluhan apabila data tersebut diperlukan lagi.

### 2.2 Pelaporan Masyarakat

Pelaporan masyarakat merupakan salah satu upaya pemerintah dalam memberikan akses kepada masyarakat untuk terlibat langsung dalam setiap perubahan ke arah yang lebih baik dan terencana. Untuk itu, perlu adanya sistem yang dapat mengelola pelaporan masyarakat dengan baik. Pelaporan yang dikelola dengan baik akan mendatangkan beberapa manfaat, seperti (BAPPENAS, 2010):

1. Pemerintah akan mengetahui kelemahan dan kekurangan dalam pelayanannya;
2. Sebagai alat introspeksi organisasi agar selalu responsif dan memerhatikan suara masyarakat;

3. Mempermudah pemerintah dalam mencari jalan keluar untuk meningkatkan mutu pelayanannya;
4. Bila segera ditangani, masyarakat merasa kepentingan dan harapannya diperhatikan;
5. Dapat menguatkan rasa kepercayaan masyarakat kepada pemerintah;
6. Penanganan komplain yang benar dapat meningkatkan kepuasan masyarakat.

Selain itu, pelaporan atau keluhan yang ditangani dengan baik juga dapat mencegah timbulnya gejolak yang mengganggu kelancaran jalannya kegiatan suatu institusi pemerintah.

### **2.2.1 Penanganan Pelaporan Masyarakat**

Penanganan pelaporan masyarakat dalam pemerintahan biasa disebut dengan “Penanganan Pengaduan”. Penanganan pengaduan merupakan kegiatan penyaluran pengaduan, pemrosesan respon atas pengaduan, umpan balik, dan laporan penanganan pengaduan. Langkah-langkah penanganan pengaduan menurut Laporan Kajian Manajemen Penanganan Pengaduan Masyarakat dalam Pelayanan Publik oleh BAPPENAS (2010), yaitu:

1. Pencatatan dan pendokumentasian;
2. Pemilihan aduan dari yang paling mendesak;
3. Mengumpulkan data dan informasi terkait dengan masalah yang diadukan;
4. Pembuatan tanggapan atau respon atas pengaduan tersebut;
5. Sampaikan respon kepada pihak pengadu untuk memperoleh umpan balik;
6. Meminta dan mencatat umpan balik dari pihak pengadu dan respon yang mereka terima;
7. Pembuatan laporan penanganan;
8. Penyampaian laporan penanganan pengaduan kepada pihak-pihak terkait.

## **2.3 Penilaian Kinerja**

### **2.3.1 Konsep Penilaian Kinerja**

Penilaian kinerja merupakan suatu proses penilaian yang dilakukan secara sistematis berdasarkan pekerjaan yang ditugaskan kepada pegawai, yang pada penelitian ini yaitu OPD. Kegiatan ini dapat memberikan umpan balik kepada pegawai tentang pelaksanaan kerja mereka, dengan tujuan menghilangkan kemerosotan prestasi kinerja (Nurmianto & Siswanto, 2006).

Penilaian kinerja memiliki beberapa manfaat, di antaranya (Messmer, 2000):

1. Menjaga pegawai tetap fokus pada tujuan dan sasaran perusahaan;
2. Memotivasi pegawai untuk meningkatkan keahlian dan pengetahuan pekerjaan agar memberikan kontribusi yang lebih;

3. Menjadi dasar obyektif bagi manajemen untuk pembuatan keputusan termasuk promosi dan kenaikan gaji;

### 2.3.2 Metode Penilaian

Pada penelitian ini, penilaian kinerja dilakukan dengan menggunakan rata-rata nilai yang didapatkan dari waktu yang dibutuhkan untuk merespon laporan masuk. Pada penilaian ini, ditentukan target maksimal yaitu 10 hari sebagai waktu maksimal untuk merespon laporan. Hal tersebut merujuk pada Undang-Undang Nomor 14 Tahun 2008 Tentang Keterbukaan Informasi Pasal 22, bahwa badan publik wajib menyampaikan pemberitahuan tertulis mengenai informasi yang diminta oleh masyarakat maksimal 10 hari kerja sejak diterimanya permintaan. Penerapan metode penilaian ini merupakan hasil wawancara dengan Dinas Kominfo Kabupaten Pekalongan.

Rumus yang digunakan pada penilaian kinerja OPD sebagai berikut (Arif, et al., 2017):

$$NA = \frac{\sum_{i=0}^n 10 - \text{selisih}_i}{n} \quad (2.1)$$

Keterangan:

1. NA = nilai akhir
2. 10 = nilai maksimal yang didapatkan OPD
3. selisih = tanggal respon – tanggal masuk laporan
4. i = indeks ke-i dari laporan yang masuk ke tiap OPD
5. n = jumlah laporan yang masuk ke tiap OPD

## 2.4 Software Development Life Cycle (SDLC)

Dalam mengembangkan perangkat lunak, terdapat siklus hidup yang berisi serangkaian langkah atau proses pengembangan perangkat lunak (*Software Development Life Cycle*). Proses tersebut direpresentasikan dalam pemodelan yang disebut *Software Process Model* atau dikenal juga dengan nama *SDLC Model*. Terdapat beberapa model dalam *SDLC Model*, masing-masing merepresentasikan pendekatan yang berbeda dalam proses pengembangan perangkat lunak (Sommerville, 2011). Pada penelitian ini, *SDLC model* yang penulis gunakan yaitu *waterfall model*.

### 2.4.1 Waterfall Model

*Waterfall model* merupakan SDLC yang memiliki pendekatan sekuensial dalam pengembangan perangkat lunak (Pressman, 2010). Pada penggunaan *waterfall model*, kebutuhan perangkat lunak harus sudah jelas dan tidak mengalami perubahan. Berikut merupakan tahapan yang ada di *waterfall model* seperti yang dapat dilihat pada Gambar 2.1 (Sommerville, 2011):

1. Definisi dan analisis kebutuhan

Analisis kebutuhan dilakukan secara intensif dengan user untuk menghasilkan spesifikasi kebutuhan agar dapat dipahami secara detail perangkat lunak seperti apa yang dibutuhkan oleh user, meliputi layanan, batasan-batasan, dan tujuan sistem.

2. Perancangan sistem dan perangkat lunak

Perancangan sistem yang dilakukan meliputi arsitektur sistem, sedangkan perancangan perangkat lunak meliputi identifikasi abstraksi dasar perangkat lunak dan relasi-relasi di dalamnya.

3. Implementasi dan pengujian unit

Tahap implementasi merupakan proses translasi perancangan perangkat lunak menjadi kumpulan program atau unit program. Sedangkan pengujian unit memverifikasi bahwa setiap unit sesuai dengan spesifikasinya.

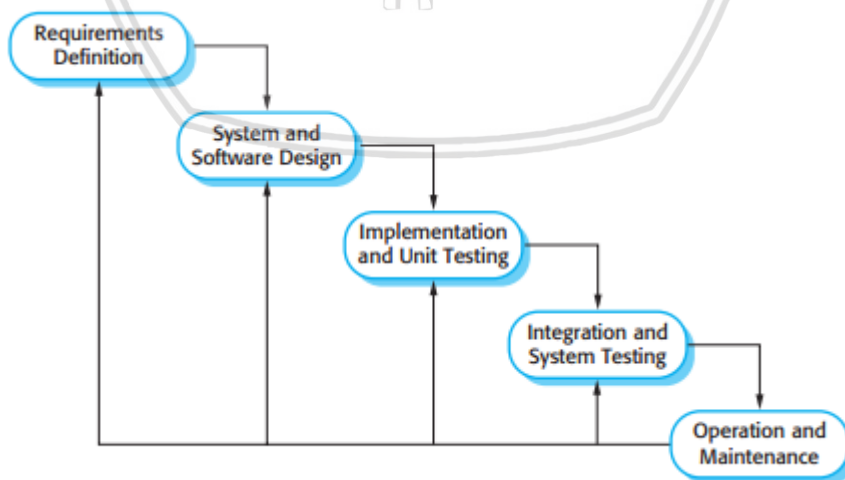
4. Pengujian integrasi dan sistem

Integrasi program diuji sebagai sistem utuh untuk memastikan bahwa program yang telah dibuat sesuai dengan kebutuhan. Setelah melakukan pengujian, perangkat lunak diberikan kepada *customer*.

5. Operasi dan pemeliharaan (tidak tentu dilakukan)

Tahap ini merupakan tahapan terpanjang. Tahap ini dapat dilakukan ketika adanya kesalahan yang muncul dan tidak terdeteksi saat pengujian, dengan mengoreksi kesalahan, meningkatkan implementasi dari unit sistem, atau mengembangkan layanan sistem sebagai sebuah kebutuhan baru.

Pada prinsipnya, setiap tahapan dalam *waterfall* model harus disetujui terlebih dahulu sebelum memasuki tahapan berikutnya. Suatu tahapan tidak dapat dimulai apabila tahapan sebelumnya belum selesai (Sommerville, 2011).



**Gambar 2.1 Ilustrasi *Waterfall Model***

Sumber: Sommerville (2011)

## 2.5 Pendekatan Berorientasi Objek

Pendekatan berorientasi objek adalah suatu strategi pengembangan perangkat lunak yang mengorganisasikan perangkat lunak sebagai kumpulan objek yang berisi data dan operasi yang diperlakukan terhadapnya (A. S. & Shalahuddin, 2016). Pendekatan berorientasi objek memiliki beberapa kelebihan (O'Docherty, 2005), yaitu:

1. Objek lebih mudah dimengerti, karena objek diturunkan dari proses bisnis yang akan diautomasi;
2. Memungkinkan komunikasi yang lebih baik antar spesialis, seperti analis, desainer, dan *programmer*;
3. Data dan proses tidak terpisah, tersimpan pada *package-package*;
4. Kode dapat digunakan ulang dengan lebih mudah.

### 2.5.1 Pemodelan

Pemodelan adalah gambaran dari realita yang disederhanakan dan direpresentasikan dalam bentuk pemetaan dengan aturan tertentu. Secara umum, pemodelan digunakan untuk merencanakan suatu hal agar kegagalan dan risiko yang mungkin terjadi dapat diminimalisasi. Pada pengembangan perangkat lunak, pemodelan ditujukan untuk mempermudah langkah berikutnya dari pengembangan sebuah sistem sehingga lebih terencana.

Pemodelan dilakukan dengan menggunakan suatu perangkat pemodelan. Perangkat pemodelan merupakan suatu model yang digunakan untuk menguraikan sistem menjadi bagian-bagian tertentu. Peran perangkat pemodelan di antaranya (A. S. & Shalahuddin, 2016):

1. Komunikasi; Perangkat pemodelan sebagai alat komunikasi antara desainer dengan analis sistem maupun *developer* dalam pengembangan sistem.
2. Eksperimentasi; Berlaku pada pengembangan sistem yang bersifat "*trial and error*".
3. Prediksi; Perangkat pemodelan meramalkan bagaimana suatu sistem akan bekerja.

Dalam metode pendekatan berorientasi objek yang cenderung tidak terlalu preskriptif, pengembang diberi beberapa pilihan dalam melakukan pemodelan (O'Docherty, 2005). Dalam penelitian ini, pemodelan yang digunakan adalah UML (*Unified Modeling Language*).

### 2.5.2 UML

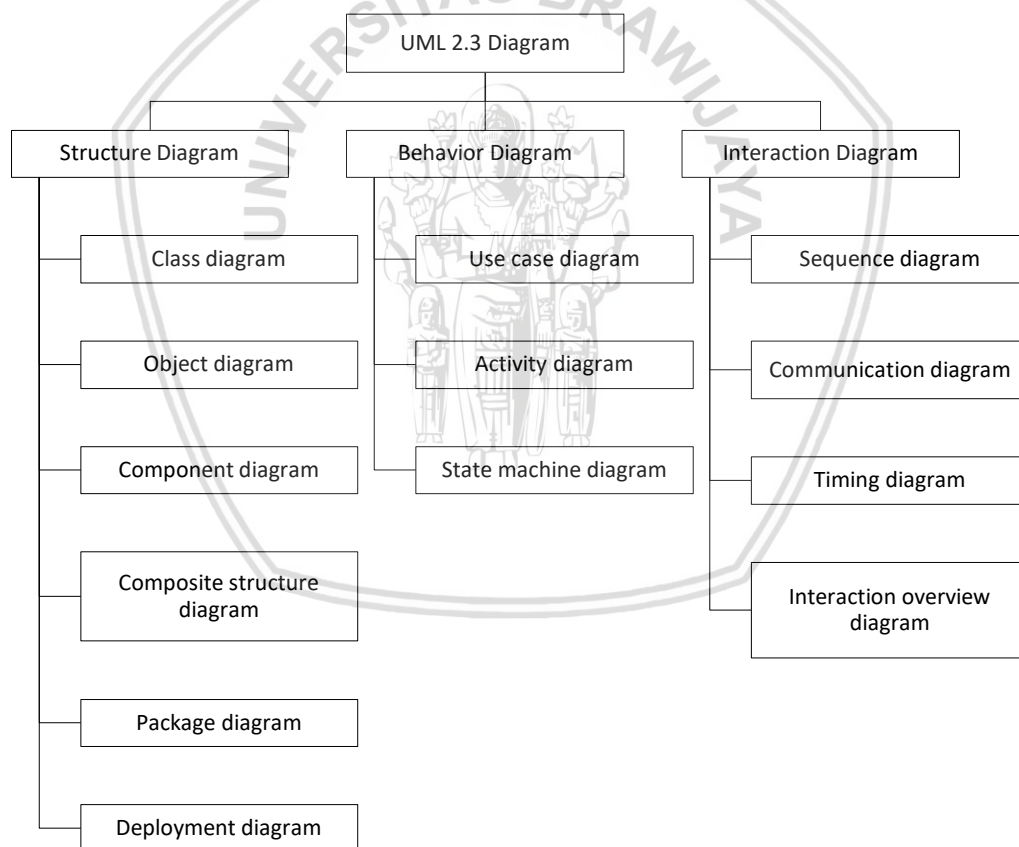
Untuk memodelkan perangkat lunak yang akan dibuat, diperlukan suatu bahasa dan standarisasi agar orang di berbagai negara mengerti pemodelan perangkat lunak tersebut (A. S. & Shalahuddin, 2016). Pada perkembangan pemrograman berorientasi objek, muncul sebuah standarisasi pemodelan untuk perangkat lunak dengan pendekatan berorientasi objek, yaitu *Unified Modeling*



*Language* (UML). UML merupakan sekelompok notasi grafik yang membantu mendeskripsikan dan merancang sistem perangkat lunak, terutama sistem yang dibangun menggunakan pendekatan berorientasi objek (Fowler, 2004).

UML diciptakan oleh Object Management Group (OMG) pada tahun 1997. Hingga saat ini, UML terbaru adalah UML 2.3 yang terdiri dari 13 macam dan dikelompokkan dalam 3 kategori, sebagaimana dapat dilihat pada Gambar 2.2. Kategori tersebut antara lain (A. S. & Shalahuddin, 2016):

1. *Structure diagram*, yaitu kumpulan diagram yang digunakan untuk menggambarkan suatu struktur statis dari sistem yang dimodelkan.
2. *Behavior diagram*, yaitu kumpulan diagram yang digunakan untuk menggambarkan kelakuan sistem atau rangkaian perubahan yang terjadi pada sebuah sistem.
3. *Interaction diagram*, yaitu kumpulan diagram yang digunakan untuk menggambarkan interaksi sistem dengan sistem lain maupun interaksi antar subsistem pada suatu sistem.



**Gambar 2.2 Diagram UML**

Sumber: A. S. & Shalahuddin (2016)

Pada penelitian ini, penulis hanya menggunakan masing-masing satu diagram dari tiap kategori, yaitu *class diagram*, *use case diagram*, dan *sequence diagram*.

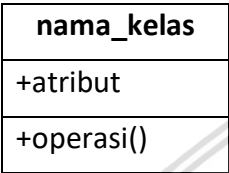



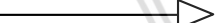




### 2.5.2.1 Class Diagram

*Class diagram* digunakan untuk menggambarkan struktur sistem dari segi pendefinisian kelas-kelas yang akan dibuat untuk membangun sistem. *Class diagram* dibuat agar *programmer* membuat kelas-kelas sesuai rancangan yang ada di dalam *class diagram* sehingga dokumentasi perancangan dan perangkat lunak sinkron (A. S. & Shalahuddin, 2016).

Berikut adalah simbol-simbol yang ada pada diagram kelas:

**Tabel 2.1 Simbol Pada Class Diagram**

Simbol	Deskripsi
Kelas 	Kelas pada struktur sistem.
Antarmuka / <i>interface</i> 	Sama dengan konsep interface dalam pemrograman berorientasi objek.
Asosiasi 	Relasi antarkelas dengan makna umum.
Asosiasi berarah 	Relasi antarkelas dengan makna kelas yang satu digunakan oleh kelas yang lain.
Generalisasi 	Relasi antarkelas dengan makna generalisasi-spesialisasi.
Kebergantungan 	Relasi antarkelas dengan makna kebergantungan antarkelas.
Agregasi 	Relasi antar kelas dengan makna semua-bagian.

Sumber: A. S. & Shalahuddin (2016)

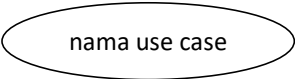


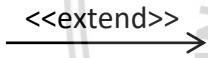

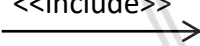
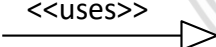
### 2.5.2.2 Use Case Diagram

*Use case diagram* digunakan untuk memodelkan kelakuan sistem yang akan dibuat (A. S. & Shalahuddin, 2016). *Use case* mendeskripsikan sebuah interaksi antara satu atau lebih aktor dengan sistem yang akan dibuat. Penamaan pada use case didefinisikan sesederhana mungkin dan dapat dipahami. Use case diagram terdiri dari dua elemen utama, yaitu aktor dan use case. Aktor adalah orang atau sistem lain yang berinteraksi dengan sistem yang akan dibuat di luar sistem itu

sendiri. Sedangkan *use case* adalah fungsionalitas yang disediakan sistem sebagai unit yang saling bertukar pesan antar unit atau aktor (A. S. & Shalahuddin, 2016).

Berikut adalah simbol-simbol yang ada pada *use case diagram*:

**Tabel 2.2 Simbol Pada Use Case Diagram**

Simbol	Deskripsi
Use case 	Fungsionalitas yang disediakan sistem.
Aktor  nama aktor	Orang atau sistem lain yang berinteraksi dengan sistem.
Asosiasi 	Komunikasi antara aktor dan <i>use case</i> .
Ekstensi/ <i>extend</i> 	Relasi <i>use case</i> tambahan ke sebuah <i>use case</i> dimana <i>use case</i> tersebut dapat berdiri sendiri walau tanpa <i>use case</i> tambahan tersebut.
Generalisasi 	Hubungan generalisasi dan spesialisasi antara dua <i>use case</i> .
Include / <i>uses</i>  	Relasi usecase tambahan ke sebuah <i>use case</i> dimana <i>use case</i> yang ditambahkan memerlukan <i>use case</i> tersebut untuk menjalankan fungsinya.




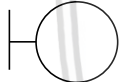


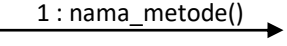
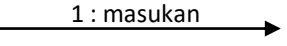
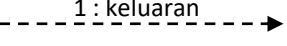
Sumber: A. S. & Shalahuddin (2016)

### 2.5.2.3 Sequence Diagram

*Sequence diagram* digunakan untuk menggambarkan kelakuan objek pada *use case* dengan mendeskripsikan waktu hidup objek dan pesan yang dikirimkan dan diterima antar objek. Pembuatan *sequence diagram* didasarkan pada skenario yang ada pada *use case*. Banyaknya *sequence diagram* yang harus digambar adalah minimal sebanyak pendefinisian *use case* yang telah dibuat (A. S. & Shalahuddin, 2016).

Berikut adalah simbol-simbol yang ada pada *sequence diagram*:

Tabel 2.3 Simbol Pada Sequence Diagram

Simbol	Deskripsi
<p>Aktor</p>  <p>nama aktor</p>	Orang atau sistem lain yang berinteraksi dengan sistem.
<p>Garis hidup / <i>lifeline</i></p> 	Menyatakan kehidupan suatu objek.
<p>Waktu aktif</p> 	Menyatakan objek dalam keadaan aktif dan berinteraksi.
<p>Objek</p>  <p>Boundary</p>  <p>Control Class</p>  <p>Entity Class</p>	<p>Menyatakan objek yang berinteraksi pesan.</p> <ul style="list-style-type: none"> <li>• Boundary: tepi dari sistem, berupa <i>user interface</i> atau alat yang berinteraksi dengan sistem atau aktor.</li> <li>• Control class: mengatur aliran informasi untuk sebuah skenario.</li> <li>• Entity class: bertanggungjawab menyimpan data atau informasi.</li> </ul>
<p>Pesan tipe call</p> 	Menyatakan suatu objek memanggil operasi/metode yang ada pada objek lain atau dirinya sendiri.
<p>Pesan tipe send</p> 	Menyatakan suatu objek mengirimkan data/masukan/informasi ke objek lain.
<p>Pesan tipe return</p> 	Menyatakan suatu objek telah menjalankan suatu operasi atau menghasilkan suatu kembalian ke objek tertentu.

Sumber: A. S. & Shalahuddin (2016)

## 2.6 Pengujian

Pengujian dilakukan untuk melihat apakah sebuah program berjalan sesuai yang dibutuhkan dan menemukan kesalahan sebelum program tersebut digunakan (Sommerville, 2011). Tujuan pengujian menurut Sommerville ada dua, antara lain:

1. Untuk mendemonstrasikan kepada pengembang dan *customer* bahwa perangkat lunak sesuai dengan kebutuhan yang telah didefinisikan sebelumnya.
2. Untuk menemukan perilaku yang salah, tidak diharapkan, atau tidak sesuai dengan spesifikasi kebutuhan pada perangkat lunak yang dikembangkan.

Adapun metode dalam pengujian perangkat lunak ada dua (Pressman, 2010), yaitu:

1. *White-Box Testing*

Metode ini juga dapat disebut *glass-box testing*, yaitu metode pengujian yang berkonsentrasi terhadap struktur program (Pressman, 2010). Dalam metode ini, pengujian dilakukan dengan memastikan bahwa semua jalur independen telah dieksekusi setidaknya sekali, menguji kondisi percabangan, menguji perulangan, dan menguji struktur data internal untuk memastikan validitasnya. Pada penelitian ini, jenis *white-box testing* yang digunakan adalah *basis path testing*. *Basis path testing* merupakan pengujian yang dilakukan untuk mengetahui jalur eksekusi yang harus dilewati berdasarkan pengukuran *cyclomatic complexity*. Sebelum menghitung *cyclomatic complexity*, alur algoritme digambarkan dalam bentuk *flow graph* (Pressman, 2010).

2. *Black-Box Testing*

Metode ini dapat disebut dengan *behavioral testing*, yaitu pengujian yang fokus pada kebutuhan fungsional perangkat lunak (Pressman, 2010). *Black-box testing* dilakukan untuk menemukan fungsi yang salah, kesalahan antarmuka, kesalahan pada struktur data atau akses *database*, kesalahan perilaku, serta kesalahan inisialisasi dan penghentian program. Pada penelitian ini, jenis *black-box testing* yang digunakan adalah *scenario-based testing* dan *equivalence partitioning*. *Scenario-based testing* adalah jenis pengujian yang fokus pada apa yang *user* lakukan, bukan apa yang sistem lakukan (Pressman, 2010). Pada jenis pengujian ini, prosedur yang harus dilakukan *user* dalam menjalankan suatu fungsionalitas diterapkan menjadi skenario uji. Sedangkan *equivalence partitioning* adalah jenis pengujian *black-box* yang membagi domain input dalam beberapa kelas data untuk mendapatkan skenario uji (Pressman, 2010).

Sedangkan untuk meningkatkan kualitas pengujian perangkat lunak, proses pengujian dibagi menjadi beberapa tahapan (Oladimeji, 2007). Tahapan yang digunakan dalam penelitian ini yaitu pengujian unit, pengujian integrasi, pengujian validasi, dan pengujian *compatibility*.

1. Pengujian Unit

Pengujian unit dilakukan pada bagian terkecil dari perangkat lunak yang dapat diuji. Dalam pemrograman terstruktur, unit merupakan fungsi atau prosedur dalam sebuah *file*. Sedangkan dalam pemrograman berorientasi objek, unit dapat berupa *method* atau kelas dalam satu *package* (Oladimeji, 2007).

## 2. Pengujian Integrasi

Pengujian integrasi dilakukan untuk memastikan bahwa antar unit berinteraksi dengan benar sebagaimana telah didefinisikan dalam perancangan perangkat lunak (Oladimeji, 2007). Pada penelitian ini, pengujian integrasi dilakukan dengan pendekatan *top-down*. *Top-down* adalah pendekatan inkremental yang menguji integrasi antar modul dengan bergerak ke bawah melalui hierarki kontrol, yang dimulai dari modul kontrol utama. Pada pendekatan ini, unit yang berada di level lebih bawah diganti dengan *stub*. *Stub* merupakan modul pengganti yang menjalankan fungsi yang mensimulasikan modul yang sebenarnya (Pressman, 2010).

## 3. Pengujian Validasi

Menurut Barry Boehm, proses validasi meliputi pertanyaan “Apakah kita membangun produk yang benar?”. Pengujian validasi dilakukan untuk memastikan bahwa perangkat lunak sesuai dengan keinginan *customer* (Sommerville, 2011).

## 4. Pengujian *Compatibility*

Pengujian *compatibility* dilakukan untuk mengetahui kemampuan aplikasi berbasis web dalam menampilkan halaman-halamannya sesuai dengan yang seharusnya pada beberapa *browser*, versi, dan *platform* (Nabil, et al., 2011). Pada penelitian ini, pengujian *compatibility* dilakukan dengan menggunakan *tool* SortSite.

## 2.7 Teknologi Pengembangan Sistem

### 2.7.1 CodeIgniter Framework

CodeIgniter dirilis pada tanggal 28 Februari 2006 oleh Rick Ellis. CodeIgniter adalah *framework* aplikasi web open-source yang digunakan untuk membangun aplikasi PHP dinamis. Karakteristik *framework* CodeIgniter di antaranya (Hustinawati, et al., 2014):

1. Fleksibel dan ringan untuk memudahkan dalam pembelajaran, modifikasi, dan integrasi dengan *library* dan *helper*.
2. Menggunakan pola MVC sehingga struktur kode lebih jelas dan sesuai dengan standar.
3. Menghasilkan alamat URL yang lebih ramah karena CodeIgniter meminimalisasi penggunaan `$_GET`.

Sebagaimana telah penulis jabarkan di atas, CodeIgniter merupakan salah satu *framework* PHP yang menerapkan pola atau arsitektur MVC. Berikut merupakan gambaran penerapan MVC pada CodeIgniter:

1. *Model*: tempat melakukan pengelolaan data dalam *database*. Di dalam model ini, umumnya terdapat fungsi untuk mengambil, mengubah, menghapus, dan menambahkan data.
2. *View*: tempat di mana kode bahasa pemrograman sisi-klien yang akan ditampilkan pada halaman web diletakkan.
3. *Controller*: mengatur hubungan antara model dan view. Di dalam *controller* umumnya terdapat alur logika dari sistem yang dibuat.

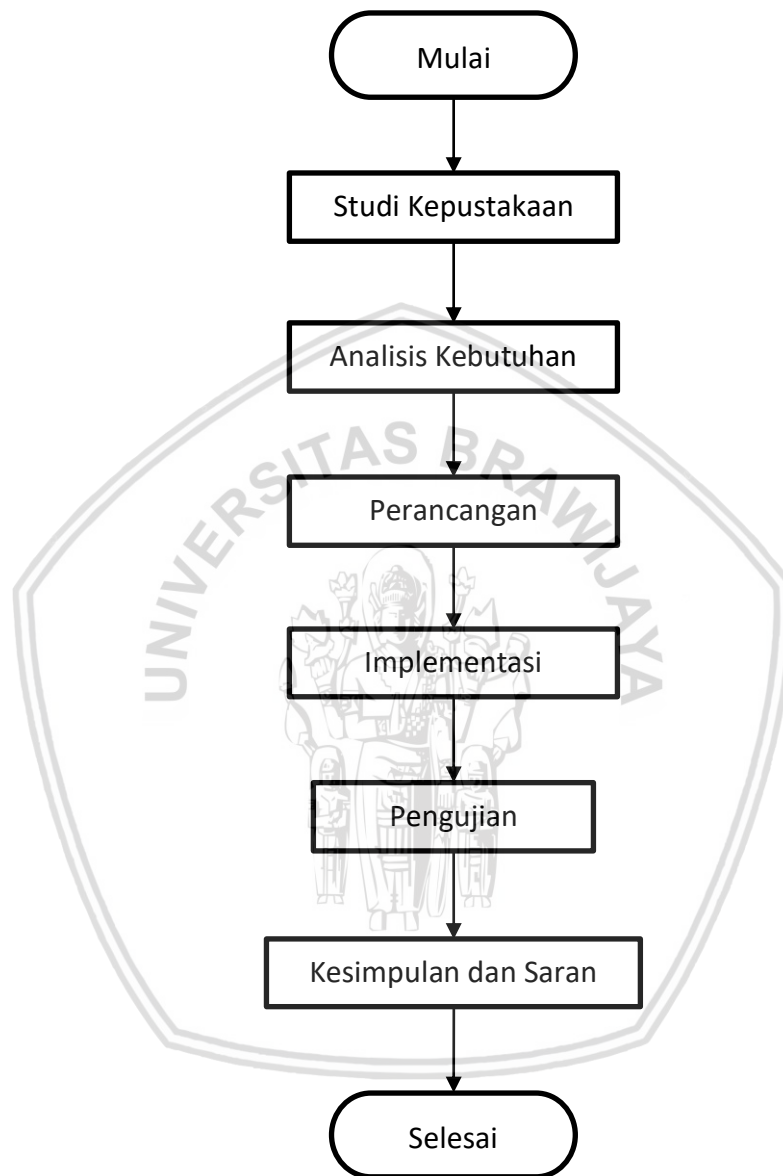
### 2.7.2 SortSite Tool

SortSite merupakan alat pengujian yang dapat memeriksa *issues* yang terdapat pada *website*, meliputi *accessibility*, *usability*, *browser compatibility*, dan *search engine optimization* (SEO). Pada penelitian ini, penulis memanfaatkan SortSite untuk menguji *compatibility* sistem aplikasi pelaporan masyarakat. Pada pengujian *compatibility*, SortSite melakukan uji pada beberapa *browser* seperti Internet Explorer, Firefox, Safari, Opera, dan Google Chrome dalam beberapa versi dan *platform*. Pengujian tersebut meliputi pengecekan lebih dari 100 *issue*, seperti penggunaan *tag* HTML dan CSS yang tidak didukung, JavaScript dan DOM yang memicu adanya *bug*, gambar yang tidak dapat ditampilkan, maupun fitur yang tidak dapat berfungsi (PowerMapper, 2015).



## BAB 3 METODOLOGI PENELITIAN

Dalam bab ini akan dijabarkan tentang metode yang dipilih serta alasan penulis memilih metode tersebut.



**Gambar 3.1 Diagram Alir Metode Penelittian**

### 3.1 Studi Kepustakaan

Studi kepustakaan digunakan untuk mempelajari teori-teori yang dijadikan bahan dasar penelitian yang dilakukan. Teori-teori tersebut meliputi:

1. Kajian Pustaka
2. Pelaporan Masyarakat

- a. Penanganan Pelaporan Masyarakat
- 3. Penilaian Kinerja
  - a. Konsep Penilaian Kinerja
  - b. Metode Penilaian
- 4. *Software Development Life Cycle (SDLC)*
  - a. *Waterfall Model*
- 5. Pendekatan Berorientasi Objek
  - a. Pemodelan
  - b. UML
    - a) *Class Diagram*
    - b) *Use Case Diagram*
    - c) *Sequence Diagram*
- 6. Pengujian
- 7. Teknologi Pengembangan Sistem
  - a. *CodeIgniter Framework*
  - b. *SortSite Tool*

### 3.2 Analisis Kebutuhan

Analisis kebutuhan merupakan kegiatan identifikasi dan spesifikasi kebutuhan sistem aplikasi pelaporan masyarakat. Adapun proses penggalian kebutuhan dalam penelitian ini dilakukan dengan wawancara. Responden penelitian yang diwawancarai yaitu Dinas Komunikasi dan Informatika (Kominfo) Kabupaten Pekalongan. Hasil wawancara tersebut dapat dilihat pada Lampiran A.

Dalam tahap ini, dilakukan pemodelan terhadap analisis data dan pemodelan terhadap kebutuhan yang telah didefinisikan. Pemodelan terhadap analisis data dilakukan dengan menggunakan ERD (*Entity Relationship Diagram*). Pemodelan ERD dilakukan untuk menjelaskan hubungan antar data dalam basis data berdasarkan objek-objek yang ada yang dihubungkan dengan suatu relasi. Sedangkan pemodelan kebutuhan dilakukan dengan menggunakan *use case diagram*. Pemodelan menggunakan *use case diagram* dilakukan untuk menjelaskan kebutuhan fungsional sistem dan aktor yang berinteraksi dengan sistem. Selanjutnya, alur kerja sistem dijelaskan pada *use case scenario* berdasarkan *use case diagram* yang telah dibuat. Selain itu, pada tahap ini juga akan dijelaskan mengenai kebutuhan non-fungsional sistem.

### 3.3 Perancangan

Setelah semua kebutuhan untuk mengembangkan sistem aplikasi pelaporan masyarakat telah terdefinisi, proses selanjutnya yaitu tahap perancangan.

Perancangan pada penelitian ini meliputi perancangan data dan perancangan sistem. Metode perancangan data digunakan untuk merancang struktur data yang akan digunakan dalam pengembangan sistem. Perancangan data ini merupakan representasi dari data-data yang dibutuhkan di dalam sistem aplikasi pelaporan masyarakat dan dimasukkan ke dalam tabel-tabel. Pemodelan data yang digunakan dalam penelitian ini adalah PDM (*Physical Data Model*). Pada PDM, data dimodelkan dalam bentuk tabel-tabel dan diidentifikasi tipe data serta hubungan antar datanya. PDM merupakan bentuk fisik perancangan basis data yang sudah siap diimplementasikan ke dalam DBMS (*Database Management System*). Sedangkan perancangan sistem pada penelitian ini dilakukan menggunakan pendekatan berorientasi objek, yaitu menggunakan diagram UML. Diagram UML yang akan digunakan pada perancangan sistem ini yaitu *sequence diagram*, dan *class diagram*. *Sequence diagram* dibuat untuk menjelaskan alur pesan antar objek, sedangkan *class diagram* dibuat untuk menjelaskan hubungan antar kelas yang ada di dalam sistem. Selain itu, pada tahap ini dilakukan pula perancangan algoritme dan perancangan antarmuka.

### 3.4 Implementasi

Tipe dari penelitian ini yaitu implementatif. Maka dari itu, setelah selesai melakukan perancangan sistem, dilanjutkan dengan melakukan tahap implementasi. Pada tahap implementasi, perancangan sistem yang telah dibuat diubah menjadi bahasa pemrograman. Bahasa pemrograman yang digunakan oleh penulis adalah PHP dengan menggunakan metode OOP (*Object-Oriented Programming*). *Framework* PHP yang digunakan penulis adalah CodeIgniter 3.1.6. Sedangkan untuk menuliskan kode program, penulis menggunakan Notepad++.

### 3.5 Pengujian

Pengujian dilakukan untuk menguji apakah perangkat lunak yang dikembangkan sudah layak untuk digunakan oleh pengguna. Selain itu, pengujian juga digunakan untuk mengetahui apakah aplikasi telah memenuhi semua kebutuhan yang didefinisikan sebelumnya. Pengujian dilakukan dengan menggunakan metode *white-box testing* dan *black-box testing*. Sedangkan tahapan pengujian yang dilakukan yaitu pengujian unit, pengujian integrasi, pengujian validasi, dan pengujian *compatibility*.

### 3.6 Kesimpulan dan Saran

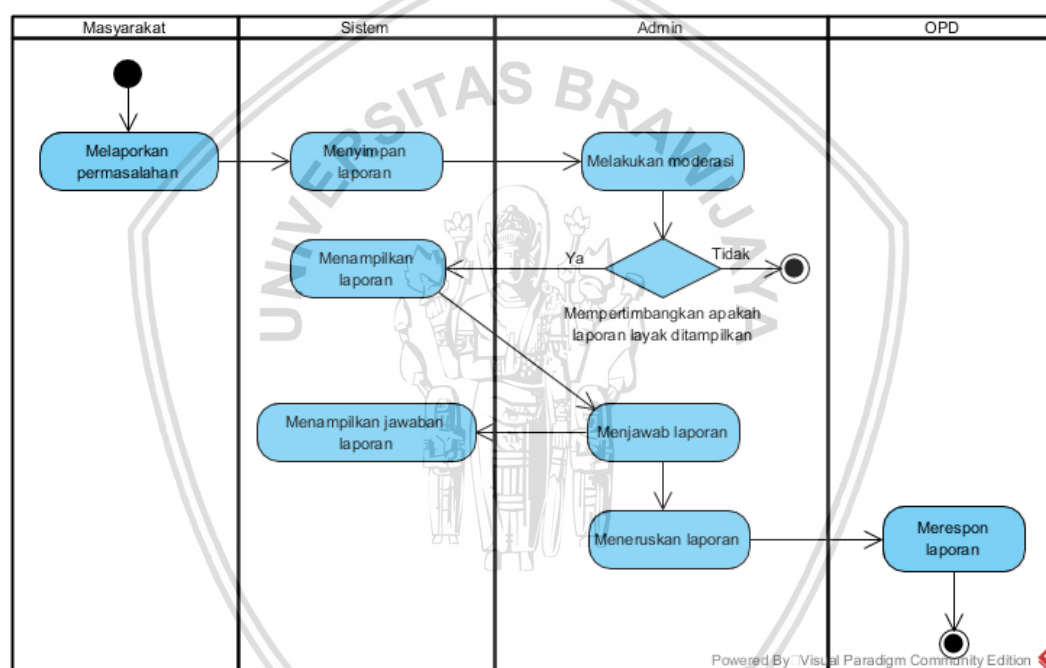
Kesimpulan dapat diambil setelah tahap perancangan, implementasi, pengujian dan analisis hasil selesai dilakukan, di mana seluruh tahap pengembangan perangkat lunak dijadikan dasar untuk melakukan pengambilan kesimpulan. Kesimpulan yang diambil nantinya akan menjawab rumusan masalah pada Bab 1.2. Selain itu, dari kesimpulan tersebut dapat ditarik saran-saran untuk penelitian dan pengembangan perangkat lunak selanjutnya.

## BAB 4 ANALISIS KEBUTUHAN

Analisis kebutuhan merupakan tahap pertama dalam pengembangan perangkat lunak. Di dalam bab ini, dijelaskan mengenai penanganan pelaporan masyarakat di Kabupaten Pekalongan saat ini, gambaran umum sistem, identifikasi aktor dalam sistem yang akan dikembangkan, kebutuhan apa saja yang terdapat pada sistem aplikasi pelaporan masyarakat, pemodelan analisis data, dan pemodelan kebutuhan.

### 4.1 Penanganan Pelaporan Masyarakat di Kabupaten Pekalongan

Berdasarkan hasil wawancara penulis dengan Dinas Kominfo Kabupaten Pekalongan, didapatkan alur aktivitas sistem pelaporan masyarakat di Kabupaten Pekalongan yang dapat dilihat pada Gambar 4.1 di bawah ini:



**Gambar 4.1 Diagram Aktivitas Pelaporan Masyarakat Saat Ini**

Pada Gambar 4.2 menunjukkan bahwa pihak yang terlibat adalah masyarakat, sistem, admin, dan OPD. Diagram aktivitas pelaporan masyarakat diinisialisasi dengan masyarakat yang melaporkan permasalahannya melalui *website* Pemerintah Kabupaten Pekalongan. Lalu, laporan akan disimpan oleh sistem. Selanjutnya, laporan yang masuk tersebut dimoderasi oleh admin. Tujuan adanya moderasi ini yaitu untuk menyaring laporan berdasarkan tata bahasa dan konteks laporan serta menentukan laporan tersebut layak ditampilkan atau tidak. Apabila laporan telah lolos moderasi, maka sistem akan menampilkan laporan. Sebaliknya, apabila laporan tidak lolos, maka laporan tidak akan ditampilkan dan proses diselesaikan. Setelah ditampilkan, admin akan menjawab laporan. Berdasarkan hasil wawancara, admin tidak memiliki kewenangan untuk menjawab laporan

secara teknis, sehingga jawaban tersebut hanya berupa tanggapan atas dikirimnya aduan dari masyarakat.

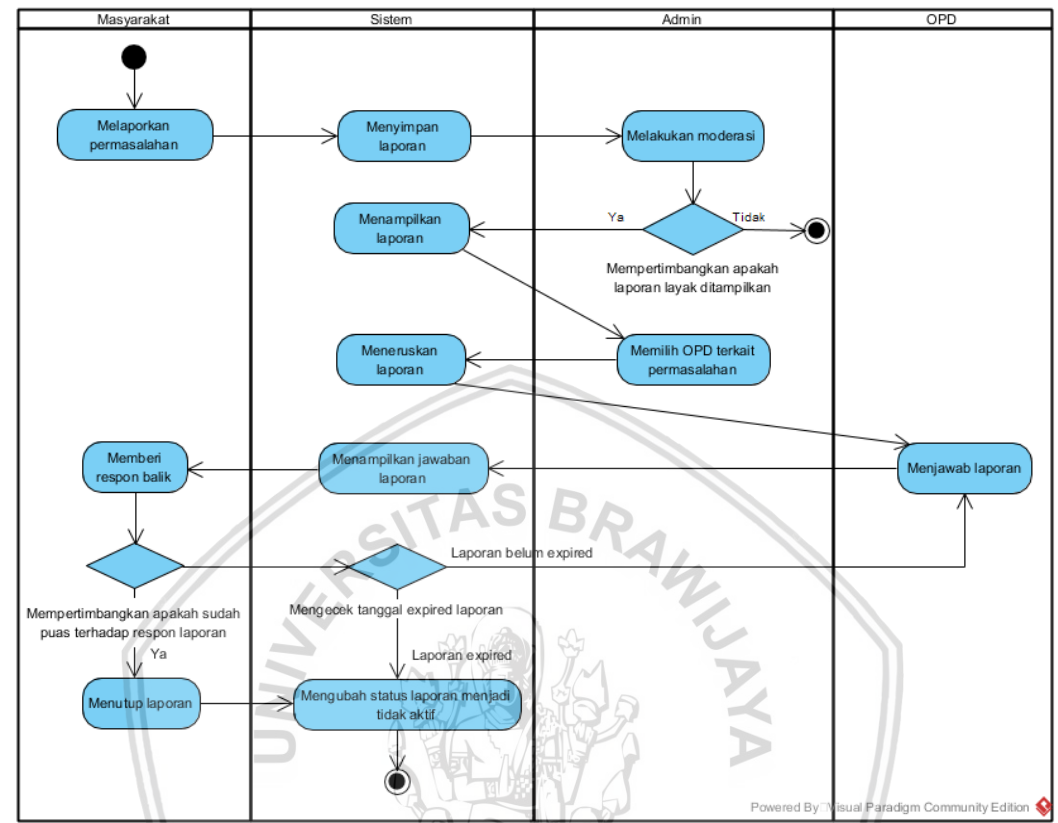
Setelah dilakukan tahap menjawab laporan, admin akan meneruskan laporan kepada OPD secara langsung. Artinya, laporan tersebut diteruskan atau didistribusikan secara manual dan tidak terintegrasi dengan sistem, yaitu dengan cara bertemu langsung atau melalui telepon. Selain manual, pendistribusian laporan yang dilakukan secara kolektif sesuai dengan permasalahannya, sehingga menimbulkan masalah pada proses penanganan yang lama serta menyebabkan masyarakat tidak mengetahui tindak lanjut dari laporan yang disampaikan. Selain itu, pada pelaksanaannya selama ini, tidak semua laporan yang ditampilkan dijawab oleh admin, sedangkan berdasarkan Undang-Undang Nomor 14 Tahun 2008 Tentang Keterbukaan Informasi Pasal 22, pemerintah wajib menyampaikan pemberitahuan tertulis mengenai informasi yang diminta oleh masyarakat maksimal 10 hari kerja sejak diterimanya permintaan.

## 4.2 Gambaran Umum Sistem

Sistem yang akan dikembangkan dalam penelitian ini yaitu sistem aplikasi pelaporan masyarakat berbasis web, yang merupakan media baru bagi masyarakat untuk menyampaikan laporan terkait permasalahan yang ditemui kepada Pemerintah Kabupaten Pekalongan. Secara umum, alur aktivitas sistem pelaporan masyarakat yang ditawarkan dijelaskan pada Gambar 4.2.

Pihak yang terlibat pada diagram aktivitas pelaporan masyarakat yang ditawarkan sama dengan diagram aktivitas saat ini, yaitu masyarakat, sistem, admin, dan OPD. Untuk dapat melaporkan permasalahan, masyarakat harus terlebih dahulu membuat akun, kemudian *login* ke sistem. Kemudian, masyarakat mengisi formulir pelaporan yang telah disediakan, serta mengunggah *file* pendukung apabila diperlukan. Maka sistem akan menyimpan laporan yang masuk. Selanjutnya admin akan melakukan moderasi laporan yang masuk untuk menentukan apakah laporan layak ditampilkan atau tidak. Laporan yang layak ditampilkan selanjutnya diteruskan kepada OPD yang terkait dengan permasalahannya oleh admin melalui sistem, di mana masing-masing OPD memiliki akun untuk masuk ke sistem sehingga OPD terkait dapat langsung menjawab dan menindaklanjuti laporan tersebut. Sistem akan menampilkan jawaban laporan dari OPD sehingga masyarakat dapat melihatnya. Setelah itu, masyarakat dapat memberikan respon balik terhadap jawaban laporan yang telah disampaikan oleh OPD. Apabila masyarakat telah merasa puas terhadap jawaban laporan, maka masyarakat dapat menutup laporan. Lalu, sistem akan mengubah status laporan menjadi tidak aktif. Sistem juga akan mengubah status laporan menjadi tidak aktif ketika mencapai tanggal *expired* laporan, yaitu 10 hari setelah respon diberikan. Penentuan 10 hari mengacu pada ketentuan pemerintah yang diwajibkan untuk memberikan respon kepada masyarakat maksimal 10 hari, sehingga hal tersebut diberlakukan pula bagi masyarakat untuk memberikan respon balik atau menutup laporan. Masyarakat juga dapat melihat penilaian kinerja masing-masing OPD dalam hal merespon laporan masyarakat yang masuk

ke dalam sistem dan rekapitulasi laporan yang masuk ke masing-masing OPD. Sistem ini diharapkan dapat menjadi solusi permasalahan dari sistem pelaporan masyarakat yang saat ini berlaku di Kabupaten Pekalongan.



Gambar 4.2 Diagram Aktivitas Pelaporan Masyarakat Yang Ditawarkan

4.3 Identifikasi Aktor

Aktor yang diidentifikasi merupakan hasil adopsi dari pihak-pihak yang terlibat dalam proses bisnis pelaporan masyarakat. Di dalam sistem aplikasi pelaporan masyarakat, aktor dibedakan menjadi empat macam yang memiliki peranannya masing-masing.

Tabel 4.1 Identifikasi Aktor

No.	Nama Aktor	Deskripsi
1.	User	User merupakan aktor yang dapat menggunakan sistem. User dapat mencari laporan, melihat penilaian kinerja tiap OPD, melihat rekapitulasi laporan, login, dan mendaftar menjadi member.
2.	Member	Member mempunyai hubungan generalisasi dengan user, karena member juga termasuk user. Member merupakan user yang dapat melaporkan permasalahan, melihat riwayat



		laporan, memberi respon balik, menutup laporan, dan <i>logout</i> .
3.	Admin	Admin mempunyai hubungan generalisasi dengan <i>user</i> , karena admin juga termasuk <i>user</i> . Admin merupakan <i>user</i> dari pihak Dinas Kominfo yang dapat melakukan moderasi laporan, meneruskan laporan, menjawab laporan, melihat OPD, menambah OPD, menghapus OPD, mengubah OPD, dan <i>logout</i> .
4.	OPD	OPD mempunyai hubungan generalisasi dengan <i>user</i> , karena OPD juga termasuk <i>user</i> . OPD merupakan <i>user</i> yang dapat menjawab laporan dan <i>logout</i> .

#### 4.4 Identifikasi Kebutuhan

Kebutuhan merupakan suatu kondisi atau kemampuan yang harus dipenuhi atau dimiliki oleh suatu sistem. Dalam sistem ini, kebutuhan yang diidentifikasi terdiri dari kebutuhan fungsional dan kebutuhan non-fungsional.

##### 4.4.1 Kebutuhan Fungsional

Kebutuhan fungsional sistem aplikasi pelaporan masyarakat didapatkan dari hasil wawancara dengan Dinas Kominfo Kabupaten Pekalongan. Setiap kebutuhan diberikan kode WADUL\_FXX. WADUL merupakan nama dari sistem yang akan dibuat, F menunjukkan kode kebutuhan fungsional, XX menunjukkan nomor kebutuhan utama.

**Tabel 4.2 Kebutuhan Fungsional**

No.	Kode Kebutuhan	Deskripsi Kebutuhan	Nama Use Case
1.	WADUL_F01	Sistem harus mampu menambahkan member baru.	Mendaftar
2.	WADUL_F02	Sistem harus mampu mengotentikasi dan mengizinkan user untuk mengakses sistem sesuai dengan otoritasnya.	<i>Login</i>
3.	WADUL_F03	Sistem harus mampu menampilkan laporan permasalahan yang masuk ke dalam sistem.	Mencari laporan

4.	WADUL_F04	Sistem harus mampu menampilkan status tahapan laporan.	Melihat status tahapan laporan
5.	WADUL_F05	Sistem harus mampu menampilkan jawaban laporan.	Melihat jawaban laporan
6.	WADUL_F06	Sistem harus mampu menampilkan statistik kinerja masing-masing OPD.	Melihat statistik kinerja OPD
7.	WADUL_F07	Sistem harus mampu mengubah bulan dan tahun statistik.	Mengubah bulan dan tahun statistik
8.	WADUL_F08	Sistem harus mampu menampilkan rekapitulasi data laporan yang diterima oleh tiap OPD.	Melihat rekapitulasi laporan
9.	WADUL_F09	Sistem harus mampu mengubah bulan dan tahun rekapitulasi.	Mengubah bulan dan tahun rekapitulasi
10.	WADUL_F10	Sistem harus mampu menyimpan data laporan yang hanya dapat diakses oleh <i>member</i> .	Melaporkan permasalahan
11.	WADUL_F11	Sistem harus mampu menampilkan riwayat laporan yang telah dikirimkan oleh <i>member</i> .	Melihat riwayat laporan
12.	WADUL_F12	Sistem harus mampu menyimpan respon balik terhadap jawaban laporan yang hanya dapat diakses oleh <i>member</i> .	Memberi respon balik
13.	WADUL_F13	Sistem harus mampu melakukan fungsi menutup laporan yang hanya dapat diakses oleh <i>member</i> .	Menutup laporan
14.	WADUL_F14	Sistem harus mampu menyimpan jawaban laporan yang hanya dapat diakses oleh OPD atau admin.	Menjawab laporan

15.	WADUL_F15	Sistem harus mampu menampilkan daftar OPD yang hanya dapat diakses oleh admin.	Melihat OPD
16.	WADUL_F16	Sistem harus mampu menambah OPD yang hanya dapat diakses oleh admin.	Menambah OPD
17.	WADUL_F17	Sistem harus mampu menghapus OPD yang hanya dapat diakses oleh admin.	Menghapus OPD
18.	WADUL_F18	Sistem harus mampu mengubah data OPD yang hanya dapat diakses oleh admin.	Mengubah OPD
19.	WADUL_F19	Sistem harus mampu melakukan moderasi laporan yang hanya dapat diakses oleh admin.	Melakukan moderasi laporan
20.	WADUL_F20	Sistem harus mampu meneruskan laporan yang hanya dapat diakses oleh admin.	Meneruskan laporan
21.	WADUL_F21	Sistem harus mampu mengeluarkan member, admin, maupun OPD dari hak aksesnya dan mengembalikan status aktor menjadi <i>user</i> .	<i>Logout</i>

#### 4.4.1.1 Spesifikasi Kebutuhan Fungsional

Setiap spesifikasi kebutuhan diberikan kode WADUL\_FXX\_YY. WADUL merupakan nama dari sistem yang akan dibuat, F menunjukkan kode kebutuhan fungsional, XX menunjukkan nomor kebutuhan utama, dan YY menunjukkan nomor spesifikasi dari kebutuhan utama.

**Tabel 4.3 Spesifikasi Kebutuhan Fungsional**

No.	Kode Kebutuhan	Kode Spesifikasi Kebutuhan	Spesifikasi
1.	WADUL_F01	WADUL_F01_01	Sistem harus mampu menyediakan <i>form</i> pendaftaran berupa NIK, nama lengkap, <i>username</i> , nomor <i>handphone</i> , dan <i>password</i> .

2.		WADUL_F01_02	Sistem harus mampu melakukan pengecekan NIK harus <i>unique</i> atau belum digunakan.
3.		WADUL_F01_03	Sistem harus mampu melakukan pengecekan tipe data pada <i>field</i> NIK berupa tipe data numerik.
4.		WADUL_F01_04	Sistem harus mampu melakukan pengecekan jumlah karakter pada <i>field</i> NIK harus berjumlah 16 karakter.
5.		WADUL_F01_05	Sistem harus mampu melakukan pengecekan <i>username</i> harus <i>unique</i> atau belum digunakan.
6.		WADUL_F01_06	Sistem harus mampu melakukan pengecekan tipe data pada <i>field</i> nomor <i>handphone</i> berupa tipe data numerik.
7.		WADUL_F01_07	Sistem harus mampu melakukan pengecekan jumlah karakter pada <i>field</i> nomor <i>handphone</i> maksimal berjumlah 12 karakter.
8.	WADUL_F02	WADUL_F02_01	Sistem harus mampu menyediakan <i>form login</i> berupa <i>username</i> dan <i>password</i> .
9.	WADUL_F03	WADUL_F03_01	Sistem harus mampu menyediakan <i>form</i> untuk memasukkan kata kunci, subjek laporan, dan nama pengirim dari laporan yang akan dicari.
10.		WADUL_F03_02	Sistem harus mampu menampilkan data laporan yang dicari berupa tanggal, nama pengirim, subjek, isi laporan, dan status laporan.
11.	WADUL_F04	WADUL_F04_01	Sistem harus mampu menampilkan status tahapan laporan ketika laporan telah diterima, diteruskan, dijawab, maupun selesai.
12.	WADUL_F05	WADUL_F05_01	Sistem harus mampu menampilkan jawaban atau

			tindaklanjut dari laporan yang disampaikan.
13.	WADUL_F06	WADUL_F06_01	Sistem harus mampu menampilkan hasil penilaian kinerja tiap OPD yang direpresentasikan dalam bentuk diagram batang.
14.	WADUL_F07	WADUL_F07_01	Sistem harus mampu menyediakan <i>dropdown</i> untuk memilih statistik yang ditampilkan berdasarkan bulan dan tahun.
15.	WADUL_F08	WADUL_F08_01	Sistem harus mampu menampilkan data rekapitulasi laporan berupa jumlah laporan yang diterima, jumlah laporan yang dapat dijawab, jumlah laporan yang tidak dapat dijawab, dan nilai yang didapat tiap bulannya.
16.	WADUL_F09	WADUL_F09_01	Sistem harus mampu menyediakan <i>dropdown</i> untuk memilih rekapitulasi yang ditampilkan berdasarkan bulan dan tahun.
17.	WADUL_F10	WADUL_F10_01	Sistem harus mampu menyediakan <i>form</i> untuk menyampaikan laporan berupa subjek laporan, isi laporan, lokasi, dan tombol <i>open file</i> untuk mengunggah gambar bila diperlukan.
18.		WADUL_F10_02	Sistem harus mampu melakukan pengecekan ekstensi <i>file</i> yang diunggah berupa ekstensi .jpg atau .png.
19.	WADUL_F12	WADUL_F12_01	Sistem harus mampu menyediakan <i>form</i> untuk memberikan respon balik terhadap jawaban laporan berupa isi respon dan tombol <i>open file</i> untuk mengunggah gambar bila diperlukan.
20.		WADUL_F12_02	Sistem harus mampu melakukan pengecekan ekstensi <i>file</i> yang

			diunggah berupa ekstensi .jpg atau .png.
21.	WADUL_F13	WADUL_F13_01	Sistem harus mampu menyediakan tombol untuk menutup laporan yang hanya bisa diakses oleh member yang menyampaikan laporan tersebut.
22.	WADUL_F14	WADUL_F14_01	Sistem harus mampu menyediakan <i>form</i> untuk menjawab laporan berupa isi jawaban, estimasi tanggal dan waktu penyelesaian masalah, dan tombol <i>open file</i> untuk mengunggah gambar bila diperlukan.
23.		WADUL_F14_02	Sistem harus mampu melakukan pengecekan ekstensi <i>file</i> yang diunggah berupa ekstensi .jpg atau .png.
24.	WADUL_F15	WADUL_F15_01	Sistem harus mampu menampilkan data OPD berupa <i>username</i> , nama OPD, <i>password</i> , NIK, dan nomor <i>handphone</i> .
25.	WADUL_F16	WADUL_F16_01	Sistem harus mampu menyediakan <i>form</i> untuk menambah OPD berupa <i>username</i> , nama OPD, NIK, nomor <i>handphone</i> , dan <i>password</i> .
26.	WADUL_F17	WADUL_F17_01	Sistem harus mampu menyediakan tombol untuk menghapus OPD.
27.	WADUL_F18	WADUL_F18_01	Sistem harus mampu menyediakan <i>form</i> untuk mengubah data OPD berupa <i>username</i> , nama OPD, NIK, nomor <i>handphone</i> , dan <i>password</i> .
28.	WADUL_F19	WADUL_F19_01	Sistem harus mampu menyediakan tombol untuk melakukan moderasi dengan memilih laporan yang masuk layak dipublikasikan atau tidak.
29.	WADUL_F20	WADUL_F20_01	Sistem harus mampu menyediakan <i>checkbox</i> untuk memilih OPD yang akan menerima laporan yang diteruskan.



#### 4.4.2 Kebutuhan Non-Fungsional

Kebutuhan non-fungsional sistem aplikasi pelaporan masyarakat didapatkan dari hasil wawancara dengan Dinas Kominfo Kabupaten Pekalongan. Setiap kebutuhan diberikan kode WADUL\_NFXX. WADUL merupakan nama dari sistem yang akan dibuat, NF merupakan kode kebutuhan non-fungsional, dan XX menunjukkan nomor kebutuhan non-fungsional.

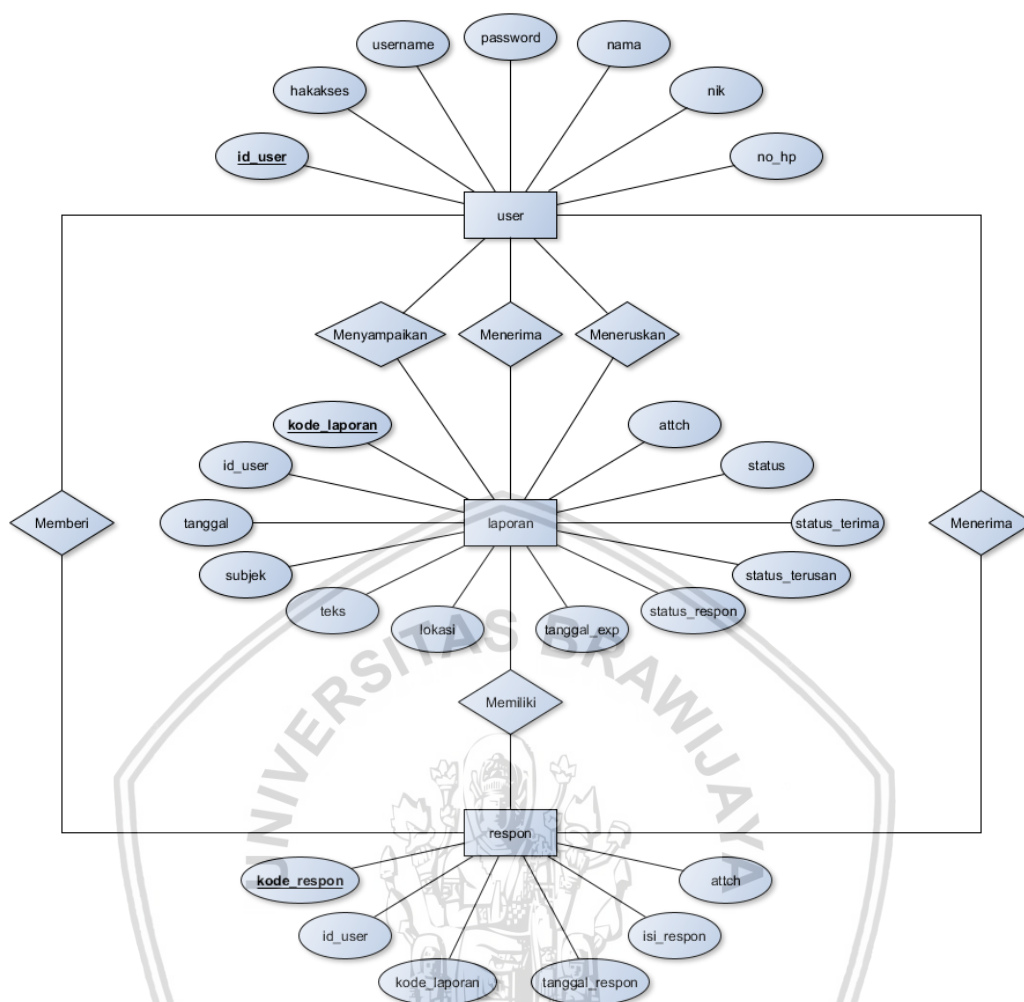
**Tabel 4.4 Kebutuhan Non-Fungsional**

No.	Kode Kebutuhan	Parameter	Deskripsi
1.	WADUL_NF01	<i>Compatibility</i>	Sistem harus dapat diakses melalui beberapa jenis <i>web browser</i> seperti Internet Explorer, Mozilla Firefox, Google Chrome, dan Safari dalam beberapa versi dan <i>platform</i> , seperti <i>desktop</i> , iOS, dan Android.

#### 4.5 Analisis Data

Analisis data dilakukan untuk mengidentifikasi entitas yang dibutuhkan dalam pengembangan sistem aplikasi pelaporan masyarakat dan relasi antar entitas yang terdapat di dalamnya. Analisis data dimodelkan dalam bentuk *Entity Relationship Diagram* (ERD).

Gambar 4.3 merupakan *entity relationship diagram* (ERD) dari sistem aplikasi pelaporan masyarakat yang menunjukkan hubungan antar entitas dalam sistem. Dalam sistem aplikasi pelaporan masyarakat, terdapat tiga entitas, yaitu *user*, laporan, dan respon yang memiliki relasi antar entitas. Entitas *user* memiliki relasi menyampaikan, menerima, dan meneruskan dengan entitas laporan serta relasi memberi dan menerima dengan entitas respon. Entitas laporan memiliki relasi memiliki dengan entitas respon.



**Gambar 4.3 Entity Relationship Diagram (ERD)**

#### 4.6 Use Case Diagram

*Use case diagram* merupakan diagram yang digunakan untuk memodelkan kelakuan sistem yang akan dibuat. *Use case* mendeskripsikan sebuah interaksi antara satu atau lebih aktor dengan sistem. *Use case diagram* pada Gambar 4.4 dibuat berdasarkan kebutuhan fungsional yang telah didefinisikan pada Tabel 4.2. *Use case diagram* sistem pelaporan masyarakat terdiri dari 21 *use case* dan 4 aktor. Aktor *User* merupakan aktor induk dari aktor yang lain, yaitu *Member*, *Admin*, dan *OPD*.



Gambar 4.4 Use Case Diagram

4.7 Use Case Scenario

Use case scenario merupakan penjelasan alur pada tiap use case yang telah didefinisikan. Use case scenario yang akan dibuat menggunakan tabel yang terdiri dari kode kebutuhan, actor, objective, pre-condition, main flow, alternative flow, dan post-condition. Jumlah use case scenario yang dijelaskan sesuai dengan jumlah use case yang telah digambarkan, yaitu 21 use case.

4.7.1 Mendaftar

Tabel 4.5 Use Case Scenario Mendaftar

Mendaftar	
Kode Kebutuhan	WADUL_F01
Actor	User

<b>Objective</b>	Menambahkan member baru.
<b>Pre-Condition</b>	Aktor telah mengakses halaman daftar.
<b>Main Flow</b>	<ol style="list-style-type: none"> <li>1. Aktor memasukkan nama lengkap, NIK, nomor <i>handphone</i>, <i>username</i>, dan <i>password</i> pada <i>form</i> daftar, lalu memilih fungsi daftar.</li> <li>2. Sistem menyimpan data <i>user</i>, memberi notifikasi bahwa <i>user</i> telah berhasil mendaftar, dan menampilkan halaman utama member.</li> </ol>
<b>Alternative Flow</b>	<ol style="list-style-type: none"> <li>1. Jika pada saat aktor memilih fungsi daftar namun ada <i>field</i> yang kosong, maka sistem akan memberikan peringatan untuk mengisi <i>field</i> yang kosong tersebut.</li> <li>2. Jika pada saat aktor memilih fungsi daftar namun NIK yang dimasukkan sudah terdaftar, maka sistem akan memberikan peringatan bahwa NIK sudah terdaftar.</li> <li>3. Jika pada saat aktor memilih fungsi daftar namun NIK yang dimasukkan bukan bertipe data numerik, maka sistem akan memberikan peringatan bahwa NIK hanya boleh diisi angka.</li> <li>4. Jika pada saat aktor memilih fungsi daftar namun NIK yang dimasukkan berjumlah kurang dari 16 karakter, maka sistem akan memberikan peringatan bahwa NIK kurang dari 16 karakter.</li> <li>5. Jika pada saat aktor memilih fungsi daftar namun NIK yang dimasukkan berjumlah lebih dari 16 karakter, maka sistem akan memberikan peringatan bahwa NIK lebih dari 16 karakter.</li> <li>6. Jika pada saat aktor memilih fungsi daftar namun <i>username</i> yang dimasukkan sudah digunakan, maka sistem akan memberikan peringatan bahwa <i>username</i> sudah digunakan.</li> <li>7. Jika pada saat aktor memilih fungsi daftar namun nomor <i>handphone</i> yang dimasukkan bukan bertipe data numerik, maka sistem akan memberikan peringatan bahwa nomor <i>handphone</i> hanya boleh diisi angka.</li> <li>8. Jika pada saat aktor memilih fungsi daftar namun nomor <i>handphone</i> yang dimasukkan berjumlah kurang dari 10 karakter, maka sistem akan</li> </ol>

	<p>memberikan peringatan bahwa nomor <i>handphone</i> kurang dari 10 karakter.</p> <p>9. Jika pada saat aktor memilih fungsi daftar namun nomor <i>handphone</i> yang dimasukkan berjumlah lebih dari 12 karakter, maka sistem akan memberikan peringatan bahwa nomor <i>handphone</i> lebih dari 12 karakter.</p>
<b>Post-Condition</b>	Aktor telah memiliki akun member dan dapat mengakses sistem sebagai member.

Tabel di atas merupakan *use case scenario* untuk *use case* mendaftar. Aktor yang dapat melakukan fungsional ini adalah *user*. Fungsi *use case* tersebut adalah untuk menambahkan member baru.

#### 4.7.2 Login

**Tabel 4.6 Use Case Scenario Login**

<b>Login</b>	
<b>Kode Kebutuhan</b>	WADUL_F02
<b>Actor</b>	<i>User</i>
<b>Objective</b>	Mengizinkan <i>user</i> mengakses sistem berdasarkan otoritasnya.
<b>Pre-Condition</b>	Aktor telah mengakses halaman utama.
<b>Main Flow</b>	<ol style="list-style-type: none"> <li>1. Aktor memasukkan <i>username</i> dan <i>password</i> pada <i>form</i> login, lalu memilih fungsi <i>login</i>.</li> <li>2. Sistem menyimpan data aktor, memberi notifikasi bahwa aktor telah berhasil login, dan menampilkan halaman utama berdasarkan otoritasnya.</li> </ol>
<b>Alternative Flow</b>	<ol style="list-style-type: none"> <li>1. Jika pada saat aktor memilih fungsi <i>login</i> namun ada <i>field</i> yang kosong, maka sistem akan memberikan peringatan untuk mengisi <i>field</i> yang kosong tersebut.</li> <li>2. Jika <i>username</i> yang dimasukkan salah, maka sistem akan menampilkan pesan "Username belum terdaftar!".</li> <li>3. Jika <i>password</i> yang dimasukkan salah, maka sistem akan menampilkan pesan "Password salah!".</li> </ol>
<b>Post-Condition</b>	Aktor dapat mengakses sistem berdasarkan otoritasnya.

Tabel di atas merupakan *use case scenario* untuk *use case login*. Aktor yang dapat melakukan fungsional ini adalah *user*. Fungsi *use case* tersebut adalah untuk mengizinkan *user* mengakses sistem berdasarkan otoritasnya.

#### 4.7.3 Mencari Laporan

**Tabel 4.7 Use Case Scenario Mencari Laporan**

Mencari Laporan	
Kode Kebutuhan	WADUL_F03
Actor	User
Objective	Mencari laporan yang masuk ke sistem.
Pre-Condition	Aktor telah mengakses halaman cari laporan.
Main Flow	<ol style="list-style-type: none"> <li>1. Aktor memasukkan kata pada <i>field</i> kata kunci, subjek laporan, dan nama pengirim dari laporan yang dicari, lalu memilih fungsi cari.</li> <li>2. Sistem menampilkan hasil pencarian.</li> </ol>
Alternative Flow	<ol style="list-style-type: none"> <li>1. Jika aktor tidak memasukkan kata pada <i>field</i> kata kunci, maka sistem akan memberikan peringatan untuk mengisi <i>field</i> yang kosong tersebut.</li> </ol>
Post-Condition	User berhasil mencari laporan.

Tabel di atas merupakan *use case scenario* untuk *use case* Mencari Laporan. Aktor yang dapat melakukan fungsional ini adalah *user*. Fungsi *use case* tersebut adalah untuk mencari laporan yang masuk ke sistem.

#### 4.7.4 Melihat Status Tahapan Laporan

**Tabel 4.8 Use Case Scenario Melihat Status Tahapan Laporan**

Melihat Status Tahapan Laporan	
Kode Kebutuhan	WADUL_F04
Actor	User
Objective	Melihat status tahapan laporan.
Pre-Condition	Aktor telah mengakses halaman utama atau telah mencari laporan.
Main Flow	<ol style="list-style-type: none"> <li>1. Aktor memilih laporan.</li> <li>2. Sistem menampilkan rincian laporan dan status tahapan laporannya.</li> </ol>
Alternative Flow	-
Post-Condition	Aktor berhasil status tahapan laporan.



Tabel di atas merupakan *use case scenario* untuk *use case* melihat status tahapan laporan. Aktor yang dapat melakukan fungsional ini adalah *user*. Fungsi *use case* tersebut adalah untuk melihat status tahapan laporan.

#### 4.7.5 Melihat Jawaban Laporan

**Tabel 4.9 Use Case Scenario Melihat Jawaban Laporan**

Melihat Jawaban Laporan	
<b>Kode Kebutuhan</b>	WADUL_F05
<b>Actor</b>	<i>User</i>
<b>Objective</b>	Melihat jawaban dari laporan yang masuk ke dalam sistem.
<b>Pre-Condition</b>	Aktor telah mengakses halaman utama atau telah mencari laporan.
<b>Main Flow</b>	<ol style="list-style-type: none"> <li>1. Aktor memilih laporan.</li> <li>2. Sistem menampilkan rincian laporan dan jawaban dari laporan tersebut.</li> </ol>
<b>Alternative Flow</b>	-
<b>Post-Condition</b>	<i>Member</i> berhasil melihat progres laporan yang telah disampaikan.

Tabel di atas merupakan *use case scenario* untuk *use case* melihat jawaban laporan. Aktor yang dapat melakukan fungsional ini adalah *member*. Fungsi *use case* tersebut adalah untuk melihat jawaban dari laporan yang masuk ke dalam sistem.

#### 4.7.6 Melihat Statistik Kinerja OPD

**Tabel 4.10 Use Case Scenario Melihat Statistik Kinerja OPD**

Melihat Statistik Kinerja OPD	
<b>Kode Kebutuhan</b>	WADUL_F06
<b>Actor</b>	<i>User</i>
<b>Objective</b>	Melihat statistik kinerja OPD.
<b>Pre-Condition</b>	Aktor telah mengakses halaman utama.
<b>Main Flow</b>	<ol style="list-style-type: none"> <li>1. Aktor membuka halaman statistik kinerja OPD.</li> <li>2. Sistem menampilkan statistik kinerja OPD pada bulan dan tahun saat ini.</li> </ol>
<b>Alternative Flow</b>	-
<b>Post-Condition</b>	Aktor berhasil melihat statistik kinerja OPD.

Tabel di atas merupakan *use case scenario* untuk *use case* melihat statistik kinerja OPD. Aktor yang dapat melakukan fungsional ini adalah *user*. Fungsi *use case* tersebut adalah untuk melihat statistik kinerja OPD.

#### 4.7.7 Mengubah Bulan dan Tahun Statistik

**Tabel 4.11 Use Case Scenario Mengubah Bulan dan Tahun Statistik**

Mengubah Bulan dan Tahun Statistik	
<b>Kode Kebutuhan</b>	WADUL_F07
<b>Actor</b>	<i>User</i>
<b>Objective</b>	Melihat statistik kinerja OPD pada bulan dan tahun terpilih.
<b>Pre-Condition</b>	Aktor telah mengakses halaman statistik.
<b>Main Flow</b>	<ol style="list-style-type: none"> <li>1. Aktor memilih bulan dan tahun lalu menekan fungsi pilih.</li> <li>2. Sistem menampilkan statistik kinerja OPD pada bulan dan tahun terpilih.</li> </ol>
<b>Alternative Flow</b>	<ol style="list-style-type: none"> <li>1. Jika pada saat aktor menekan fungsi pilih namun ada <i>field</i> yang kosong, maka sistem akan memberikan peringatan untuk mengisi <i>field</i> yang kosong tersebut.</li> </ol>
<b>Post-Condition</b>	Aktor berhasil melihat statistik kinerja OPD pada bulan dan tahun terpilih.

Tabel di atas merupakan *use case scenario* untuk *use case* mengubah bulan dan tahun statistik. Aktor yang dapat melakukan fungsional ini adalah *user*. Fungsi *use case* tersebut adalah untuk melihat statistik kinerja OPD pada bulan dan tahun terpilih.

#### 4.7.8 Melihat Rekapitulasi Laporan

**Tabel 4.12 Use Case Scenario Melihat Rekapitulasi Laporan**

Melihat Rekapitulasi Laporan	
<b>Kode Kebutuhan</b>	WADUL_F08
<b>Actor</b>	<i>User</i>
<b>Objective</b>	Melihat rekapitulasi laporan yang masuk ke sistem.
<b>Pre-Condition</b>	Aktor telah mengakses halaman utama utama.
<b>Main Flow</b>	<ol style="list-style-type: none"> <li>1. Aktor membuka halaman rekapitulasi laporan.</li> <li>2. Sistem menampilkan rekapitulasi laporan pada bulan dan tahun saat ini.</li> </ol>

<b>Alternative Flow</b>	-
<b>Post-Condition</b>	Aktor berhasil melihat rekapitulasi laporan.

Tabel di atas merupakan *use case scenario* untuk *use case* melihat rekapitulasi laporan. Aktor yang dapat melakukan fungsional ini adalah *user*. Fungsi *use case* tersebut adalah untuk melihat rekapitulasi laporan yang masuk ke sistem.

#### 4.7.9 Mengubah Bulan dan Tahun Rekapitulasi

Mengubah Bulan dan Tahun Rekapitulasi	
<b>Kode Kebutuhan</b>	WADUL_F09
<b>Aktor</b>	<i>User</i>
<b>Objective</b>	Melihat rekapitulasi laporan yang masuk ke sistem pada bulan dan tahun terpilih.
<b>Pre-Condition</b>	Aktor telah mengakses halaman utama rekapitulasi laporan.
<b>Main Flow</b>	<ol style="list-style-type: none"> <li>1. Aktor memilih bulan dan tahun lalu menekan fungsi pilih.</li> <li>2. Sistem menampilkan rekapitulasi laporan pada bulan dan tahun terpilih.</li> </ol>
<b>Alternative Flow</b>	<ol style="list-style-type: none"> <li>1. Jika pada saat aktor menekan fungsi pilih namun ada <i>field</i> yang kosong, maka sistem akan memberikan peringatan untuk mengisi <i>field</i> yang kosong tersebut.</li> </ol>
<b>Post-Condition</b>	Aktor berhasil melihat rekapitulasi laporan pada bulan dan tahun terpilih.

Tabel di atas merupakan *use case scenario* untuk *use case* mengubah bulan dan tahun rekapitulasi. Aktor yang dapat melakukan fungsional ini adalah *user*. Fungsi *use case* tersebut adalah untuk melihat rekapitulasi laporan yang masuk ke sistem pada bulan dan tahun terpilih.

#### 4.7.10 Melaporkan Permasalahan

**Tabel 4.13 Use Case Scenario Melaporkan Permasalahan**

Melaporkan Permasalahan	
<b>Kode Kebutuhan</b>	WADUL_F10
<b>Aktor</b>	<i>Member</i>
<b>Objective</b>	Melaporkan permasalahan kepada Pemkab Pekalongan.
<b>Pre-Condition</b>	Aktor telah mengakses halaman formulir pelaporan.

<b>Main Flow</b>	<ol style="list-style-type: none"> <li>1. Aktor mengisi <i>form</i> pelaporan dengan memasukkan subjek laporan, isi laporan yang akan disampaikan, lokasi permasalahan, dan mengunggah <i>file</i> apabila diperlukan, lalu memilih fungsi kirim.</li> <li>2. Sistem menyimpan laporan aktor dan memberi notifikasi "Data berhasil disimpan".</li> </ol>
<b>Alternative Flow</b>	<ol style="list-style-type: none"> <li>1. Jika pada saat aktor menekan fungsi kirim namun ada <i>field</i> yang kosong, maka sistem akan memberikan peringatan untuk mengisi <i>field</i> kosong tersebut.</li> <li>2. Jika pada saat aktor menekan fungsi kirim namun <i>file</i> yang diunggah bukan berekstensi .jpg atau .png, maka sistem akan menampilkan pesan gagal melaporkan permasalahan.</li> </ol>
<b>Post-Condition</b>	Aktor berhasil melaporkan permasalahan dan status tahapan laporan diperbaharui.

Tabel di atas merupakan *use case scenario* untuk *use case* melaporkan permasalahan. Aktor yang dapat melakukan fungsional ini adalah *member*. Fungsi *use case* tersebut adalah untuk melaporkan permasalahan kepada Pemkab Pekalongan.

#### 4.7.11 Melihat Riwayat Laporan

Tabel 4.14 *Use Case Scenario* Melihat Riwayat Laporan

Melihat Riwayat Laporan	
<b>Kode Kebutuhan</b>	WADUL_F11
<b>Actor</b>	<i>Member</i>
<b>Objective</b>	Melihat riwayat laporan yang telah disampaikan.
<b>Pre-Condition</b>	Aktor telah mengakses halaman utama.
<b>Main Flow</b>	<ol style="list-style-type: none"> <li>1. Aktor membuka halaman "Laporan Saya".</li> <li>2. Sistem menampilkan daftar laporan yang dikirimkan aktor.</li> </ol>
<b>Alternative Flow</b>	-
<b>Post-Condition</b>	Aktor berhasil melihat progres laporan yang telah disampaikan.

Tabel di atas merupakan *use case scenario* untuk *use case* melihat riwayat laporan. Aktor yang dapat melakukan fungsional ini adalah *member*. Fungsi *use case* tersebut adalah untuk melihat riwayat laporan yang telah disampaikan.

#### 4.7.12 Memberi Respon Balik

Tabel 4.15 *Use Case Scenario* Memberi Respon Balik

Memberi Respon Balik	
Kode Kebutuhan	WADUL_F12
Actor	Member
Objective	Memberi respon balik terhadap jawaban laporan.
Pre-Condition	Aktor telah mengakses halaman rincian laporan.
Main Flow	<ol style="list-style-type: none"> <li>1. Aktor mengisi <i>form</i> respon dengan memasukkan isi respon yang akan disampaikan dan mengunggah <i>file</i> apabila diperlukan, lalu memilih fungsi kirim.</li> <li>2. Sistem menyimpan respon balik.</li> </ol>
Alternative Flow	<ol style="list-style-type: none"> <li>1. Jika pada saat aktor menekan tombol kirim namun <i>field</i> isi respon kosong, maka sistem akan memberikan peringatan untuk mengisi <i>field</i> kosong tersebut.</li> <li>2. Jika pada saat aktor menekan fungsi kirim namun <i>file</i> yang diunggah bukan berekstensi .jpg atau .png, maka sistem akan menampilkan pesan gagal mengirimkan respon.</li> </ol>
Post-Condition	Aktor berhasil memberi respon balik.

Tabel di atas merupakan *use case scenario* untuk *use case* memberi respon balik. Aktor yang dapat melakukan fungsional ini adalah *member*. Fungsi *use case* tersebut adalah untuk memberi respon balik terhadap jawaban laporan.

#### 4.7.13 Menutup Laporan

Tabel 4.16 *Use Case Scenario* Menutup Laporan

Menutup Laporan	
Kode Kebutuhan	WADUL_F13
Actor	Aktor
Objective	Menutup laporan yang telah disampaikan.
Pre-Condition	Aktor telah mengakses halaman rincian laporan.
Main Flow	<ol style="list-style-type: none"> <li>1. Aktor memilih fungsi untuk menutup laporan.</li> <li>2. Sistem menampilkan dialog konfirmasi penutupan laporan.</li> <li>3. Aktor menyetujui penghapusan OPD.</li> </ol>

	4. Sistem mengubah status laporan menjadi “Tidak Aktif”.
<b>Alternative Flow</b>	1. Jika aktor tidak menyetujui penutupan laporan, maka sistem membatalkan penutupan.
<b>Post-Condition</b>	Aktor berhasil menutup laporan yang telah disampaikan.

Tabel di atas merupakan *use case scenario* untuk *use case* menutup laporan. Aktor yang dapat melakukan fungsional ini adalah *member*. Fungsi *use case* tersebut adalah untuk menutup laporan yang telah disampaikan.

#### 4.7.14 Menjawab Laporan

**Tabel 4.17 Use Case Scenario Menjawab Laporan**

Menjawab Laporan	
<b>Kode Kebutuhan</b>	WADUL_F14
<b>Actor</b>	OPD, Admin
<b>Objective</b>	Menjawab laporan yang disampaikan oleh <i>member</i> .
<b>Pre-Condition</b>	Aktor telah mengakses halaman rincian laporan.
<b>Main Flow</b>	<ol style="list-style-type: none"> <li>1. Aktor mengisi <i>form</i> jawaban dengan memasukkan isi laporan yang akan disampaikan, estimasi tanggal dan waktu penyelesaian masalah, dan mengunggah <i>file</i> apabila diperlukan, lalu memilih fungsi kirim.</li> <li>2. Sistem menyimpan jawaban laporan.</li> </ol>
<b>Alternative Flow</b>	<ol style="list-style-type: none"> <li>1. Jika pada saat aktor menekan tombol kirim ada <i>field</i> yang kosong, maka sistem akan memberikan peringatan untuk mengisi <i>field</i> kosong tersebut.</li> <li>2. Jika pada saat aktor menekan fungsi kirim namun <i>file</i> yang diunggah bukan berekstensi .jpg atau .png, maka sistem akan menampilkan pesan gagal mengirimkan respon.</li> </ol>
<b>Post-Condition</b>	Aktor berhasil menjawab laporan dan status tahapan laporan diperbaharui.

Tabel di atas merupakan *use case scenario* untuk *use case* menjawab laporan. Aktor yang dapat melakukan fungsional ini adalah OPD dan Admin. Fungsi *use case* tersebut adalah untuk menjawab laporan yang disampaikan oleh *member*.



#### 4.7.15 Melihat OPD

Tabel 4.18 *Use Case Scenario* Melihat OPD

Melihat OPD	
Kode Kebutuhan	WADUL_F15
Actor	Admin
Objective	Melihat daftar OPD.
Pre-Condition	Aktor telah mengakses halaman utama admin.
Main Flow	<ol style="list-style-type: none"> <li>1. Aktor membuka halaman kelola daftar OPD.</li> <li>2. Sistem menampilkan data OPD yang telah terdaftar dalam sistem.</li> </ol>
Alternative Flow	-
Post-Condition	Admin berhasil melihat daftar OPD.

Tabel di atas merupakan *use case scenario* untuk *use case* melihat OPD. Aktor yang dapat melakukan fungsional ini adalah admin. Fungsi *use case* tersebut adalah untuk melihat daftar OPD.

#### 4.7.16 Menambah OPD

Tabel 4.19 *Use Case Scenario* Menambah OPD

Menambah OPD	
Kode Kebutuhan	WADUL_F16
Actor	Admin
Objective	Menambah OPD baru.
Pre-Condition	Aktor telah mengakses halaman kelola daftar OPD.
Main Flow	<ol style="list-style-type: none"> <li>1. Aktor memilih fungsi tambah OPD.</li> <li>2. Sistem menampilkan <i>form</i> penambahan OPD.</li> <li>3. Aktor memasukkan username, nama OPD, NIK, nomor <i>handphone</i>, dan <i>password</i>, lalu memilih fungsi untuk menambahkan.</li> <li>4. Sistem menyimpan data OPD dan menampilkan halaman daftar OPD yang baru.</li> </ol>
Alternative Flow	<ol style="list-style-type: none"> <li>1. Jika pada saat aktor menekan tombol untuk menambahkan namun <i>field</i> pesan kosong, maka sistem akan memberikan peringatan untuk mengisi <i>field</i> kosong tersebut.</li> </ol>
Post-Condition	Aktor berhasil menambah OPD.

Tabel di atas merupakan *use case scenario* untuk *use case* menambah OPD. Aktor yang dapat melakukan fungsional ini adalah admin. Fungsi *use case* tersebut adalah untuk menambah OPD baru.

#### 4.7.17 Menghapus OPD

**Tabel 4.20 Use Case Scenario Menghapus OPD**

Menghapus OPD	
Kode Kebutuhan	WADUL_F17
Actor	Admin
Objective	Menghapus data OPD.
Pre-Condition	Aktor telah mengakses halaman kelola daftar OPD.
Main Flow	<ol style="list-style-type: none"> <li>1. Aktor memilih fungsi hapus OPD pada OPD yang akan dihapus.</li> <li>2. Sistem menampilkan dialog konfirmasi penghapusan OPD.</li> <li>3. Aktor menyetujui penghapusan OPD.</li> <li>4. Sistem menghapus data OPD dan menampilkan halaman daftar OPD yang baru.</li> </ol>
Alternative Flow	<ol style="list-style-type: none"> <li>1. Jika aktor tidak menyetujui penghapusan OPD, maka sistem membatalkan penghapusan.</li> </ol>
Post-Condition	Aktor berhasil menghapus OPD.

Tabel di atas merupakan *use case scenario* untuk *use case* menghapus OPD. Aktor yang dapat melakukan fungsional ini adalah admin. Fungsi *use case* tersebut adalah untuk admin menghapus OPD.

#### 4.7.18 Mengubah OPD

**Tabel 4.21 Use Case Scenario Mengubah OPD**

Mengubah OPD	
Kode Kebutuhan	WADUL_F18
Actor	Admin
Objective	Mengizinkan admin mengubah data OPD.
Pre-Condition	Aktor telah mengakses halaman daftar OPD.
Main Flow	<ol style="list-style-type: none"> <li>1. Aktor memilih fungsi ubah OPD pada OPD yang akan dihapus.</li> <li>2. Sistem menampilkan <i>form</i> ubah OPD.</li> <li>3. Aktor melakukan perubahan pada data OPD.</li> </ol>

	4. Sistem menyimpan data OPD dan menampilkan halaman daftar OPD yang baru.
<b>Alternative Flow</b>	-
<b>Post-Condition</b>	Aktor berhasil mengubah data OPD.

Tabel di atas merupakan *use case scenario* untuk *use case* mengubah OPD. Aktor yang dapat melakukan fungsional ini adalah admin. Fungsi *use case* tersebut adalah untuk mengubah data OPD.

#### 4.7.19 Melakukan Moderasi Laporan

**Tabel 4.22 Use Case Scenario Melakukan Moderasi Laporan**

Melakukan Moderasi Laporan	
<b>Kode Kebutuhan</b>	WADUL_F19
<b>Actor</b>	Admin
<b>Objective</b>	Melakukan moderasi terhadap laporan yang masuk ke sistem.
<b>Pre-Condition</b>	Aktor telah mengakses halaman moderasi laporan.
<b>Main Flow</b>	<ol style="list-style-type: none"> <li>1. Aktor memilih fungsi untuk mempublikasikan laporan.</li> <li>2. Sistem mengubah status publikasi laporan menjadi "Published".</li> </ol>
<b>Alternative Flow</b>	<ol style="list-style-type: none"> <li>1. Jika status publikasi laporan "Published" dan aktor ingin membatalkan publikasi laporan, aktor memilih fungsi untuk membatalkan publikasi laporan, maka sistem mengubah status publikasi laporan menjadi "Unpublished".</li> </ol>
<b>Post-Condition</b>	Aktor berhasil melakukan moderasi laporan.

Tabel di atas merupakan *use case scenario* untuk *use case* melakukan moderasi laporan. Aktor yang dapat melakukan fungsional ini adalah admin. Fungsi *use case* tersebut adalah untuk melakukan moderasi terhadap laporan yang masuk ke sistem.

#### 4.7.20 Meneruskan Laporan

Meneruskan Laporan	
<b>Kode Kebutuhan</b>	WADUL_F20
<b>Actor</b>	Admin
<b>Objective</b>	Meneruskan laporan yang masuk ke OPD yang terkait dengan permasalahan.

<b>Pre-Condition</b>	Aktor telah mengakses halaman moderasi laporan.
<b>Main Flow</b>	<ol style="list-style-type: none"> <li>1. Aktor memilih fungsi untuk meneruskan laporan.</li> <li>2. Sistem menampilkan daftar OPD.</li> <li>3. Aktor memilih OPD yang terkait dengan permasalahan.</li> <li>4. Sistem menyimpan data yang dipilih.</li> </ol>
<b>Alternative Flow</b>	<ol style="list-style-type: none"> <li>1. Jika pada saat menyimpan data tidak ada OPD yang dipilih, maka sistem akan menampilkan pesan "Laporan gagal diteruskan".</li> </ol>
<b>Post-Condition</b>	Aktor berhasil meneruskan laporan dan status tahapan laporan diperbaharui.

Tabel di atas merupakan *use case scenario* untuk *use case* meneruskan laporan. Aktor yang dapat melakukan fungsional ini adalah admin. Fungsi *use case* tersebut adalah untuk meneruskan laporan yang masuk ke OPD yang terkait dengan permasalahan.

#### 4.7.21 Logout

**Tabel 4.23 Use Case Scenario Logout**

<b>Logout</b>	
<b>Kode Kebutuhan</b>	WADUL_F21
<b>Aktor</b>	<i>Member</i> , Admin, dan OPD
<b>Objective</b>	Mengeluarkan member, admin, dan OPD dari otoritasnya dan mengembalikan status aktor menjadi <i>user</i> .
<b>Pre-Condition</b>	Aktor telah <i>login</i> .
<b>Main Flow</b>	<ol style="list-style-type: none"> <li>1. Aktor memilih fungsi <i>logout</i>.</li> <li>2. Sistem menampilkan halaman utama <i>user</i>.</li> </ol>
<b>Alternative Flow</b>	-
<b>Post-Condition</b>	Aktor berhasil keluar dari otoritasnya dan status aktor kembali menjadi <i>user</i> .

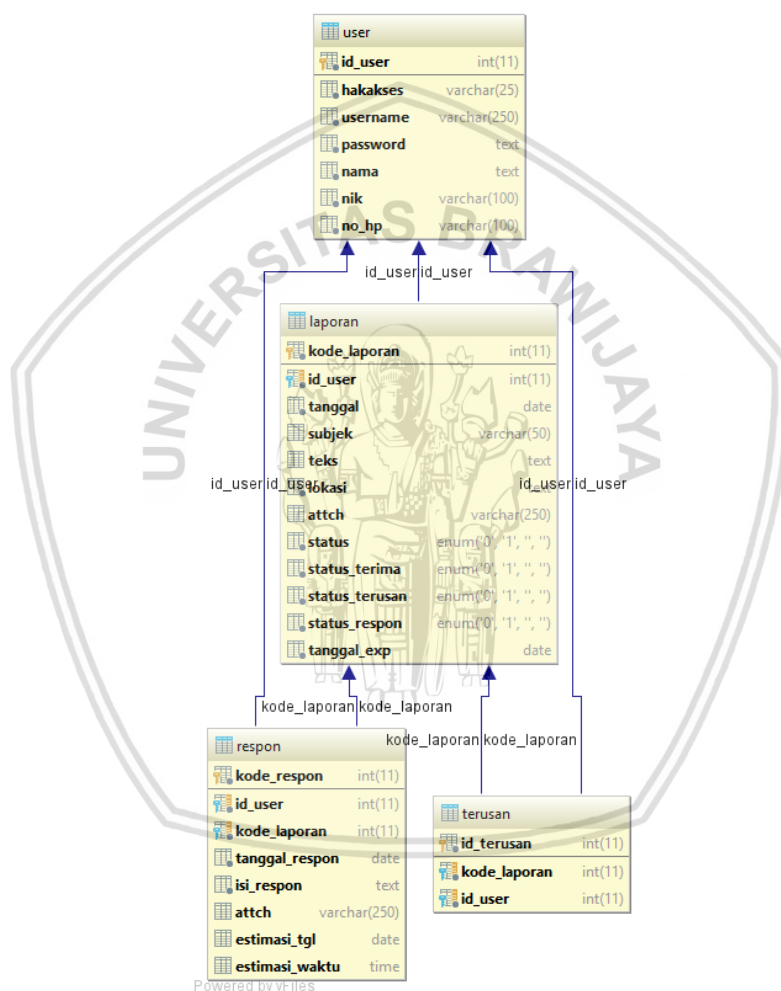
Tabel di atas merupakan *use case scenario* untuk *use case logout*. Aktor yang dapat melakukan fungsional ini adalah *member*, admin, dan OPD. Fungsi *use case* tersebut adalah mengeluarkan member, admin, dan OPD dari otoritasnya dan mengembalikan status aktor menjadi *user*.

## BAB 5 PERANCANGAN DAN IMPLEMENTASI

### 5.1 Perancangan

Pada bagian perancangan ini, akan dijabarkan mengenai perancangan yang dilakukan sebelum dilanjutkan pada tahap implementasi. Perancangan tersebut terdiri dari perancangan data, *sequence diagram*, *class diagram*, perancangan algoritme, dan perancangan antarmuka.

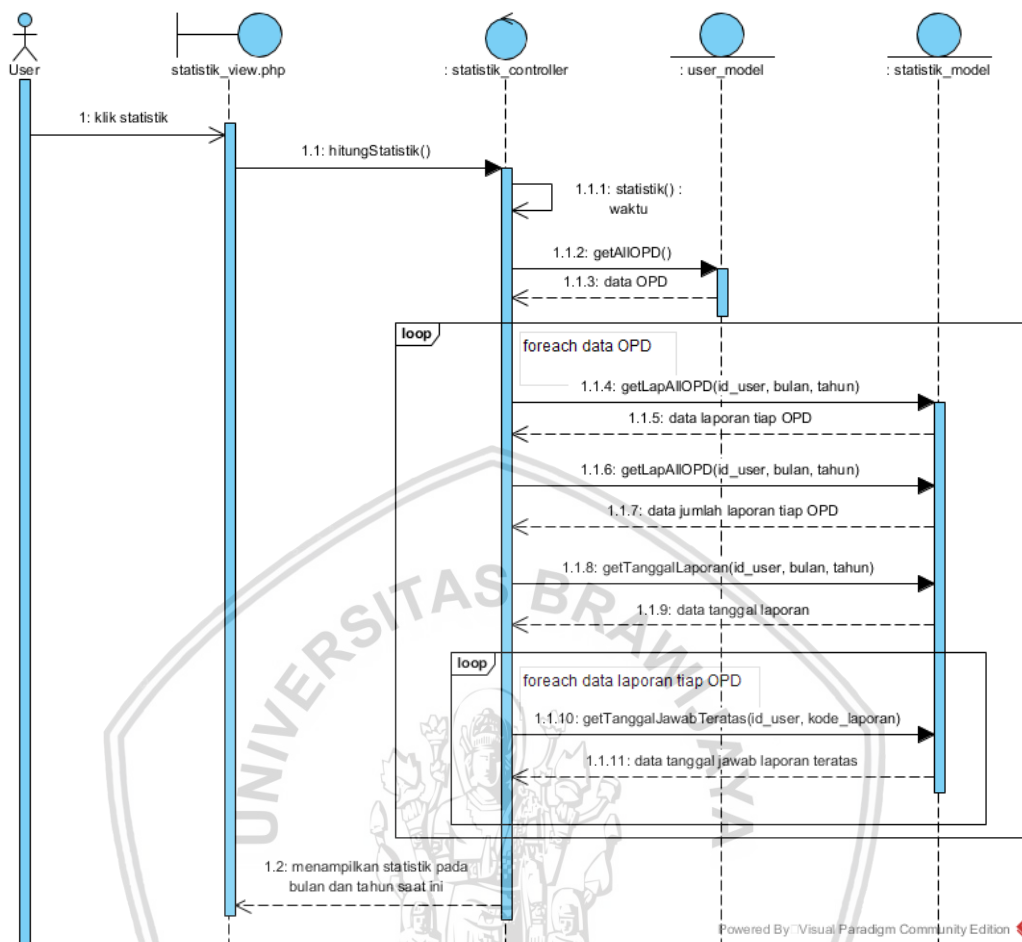
#### 5.1.1 Perancangan Data



**Gambar 5.1 Physical Data Model (PDM)**

Gambar 5.1 merupakan PDM dari basis data “wadul”. Basis data ini merupakan basis data utama yang digunakan di dalam sistem aplikasi pelaporan masyarakat. Pembuatan PDM dilakukan berdasarkan analisis data yang telah dibuat. Dalam basis data ini, terdapat empat tabel, yaitu user, laporan, respon, dan terusan.

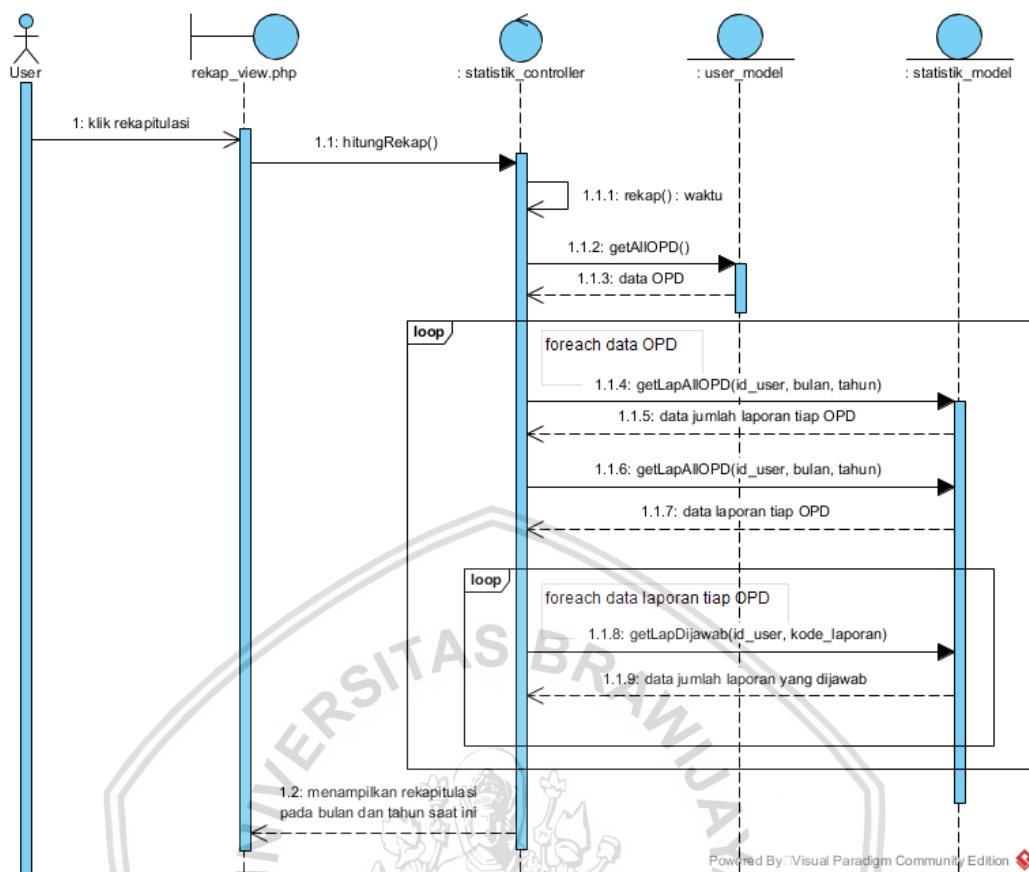
### 5.1.2 Pemodelan Sequence Diagram



Gambar 5.2 Sequence Diagram Melihat Statistik Kinerja OPD

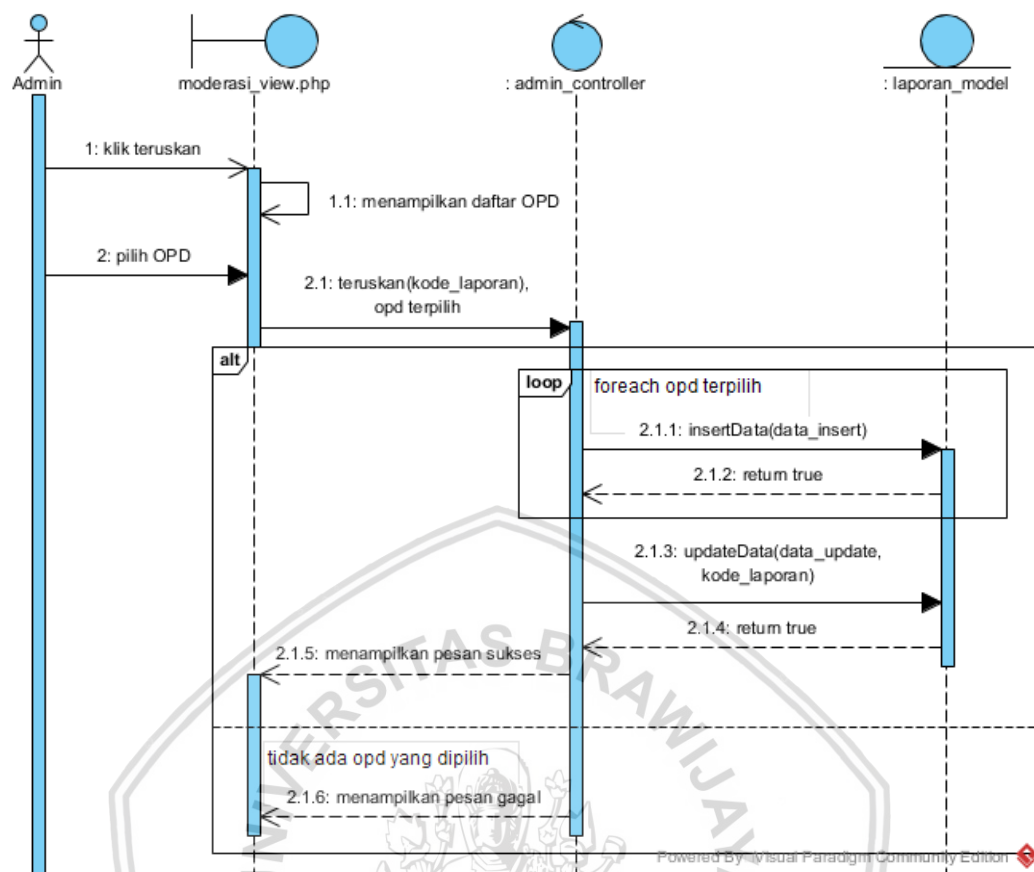
Gambar 5.2 merupakan *sequence diagram* dari melihat statistik kinerja OPD. Untuk melakukan aktivitas melihat statistik kinerja OPD, aktor mengakses halaman statistik, lalu *boundary* statistik\_view akan memanggil *method* hitungStatistik() pada *controller* statistik\_controller, maka *method* tersebut akan memanggil *method* statistik() untuk mengambil data waktu yang berisi bulan dan tahun saat ini, lalu kembali ke *method* hitungStatistik() dan memanggil *method* getAllOPD() pada *entity* user\_model untuk mendapatkan data OPD. Lalu *controller* statistik\_controller melakukan *looping* terhadap data OPD. Dalam *looping* tersebut, *controller* statistik\_controller memanggil *method* getLapAllOPD(id\_user, bulan, tahun) pada *entity* statistik\_model untuk mendapatkan data laporan tiap OPD dan data jumlah laporan tiap OPD, serta *method* getTanggalLaporan(id\_user, bulan, tahun) untuk mendapatkan data tanggal laporan. Lalu dilakukan *looping* terhadap data laporan tiap OPD, dimana *looping* tersebut memanggil *method* getTanggalJawabTeratas(id\_user, kode\_laporan) untuk mendapatkan data tanggal jawab teratas. Setelah *looping* selesai, *controller* statistik\_controller menampilkan statistik kinerja OPD pada bulan dan tahun saat ini melalui *boundary* statistik\_view.





**Gambar 5.3 Sequence Diagram Melihat Rekapitulasi Laporan**

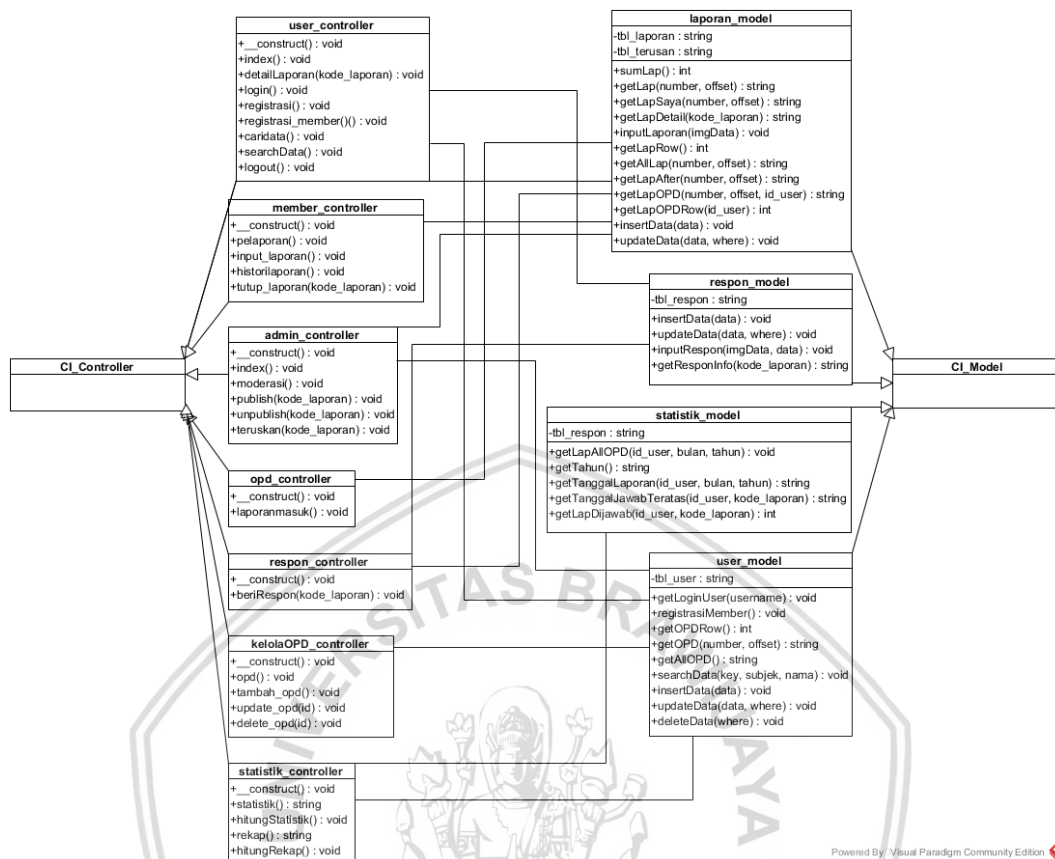
Gambar 5.3 merupakan *sequence diagram* dari melihat rekapitulasi laporan. Untuk melakukan aktivitas melihat statistik kinerja OPD, aktor mengakses halaman rekapitulasi, lalu *boundary* rekap\_view akan memanggil *method* hitungRekap() pada *controller* statistik\_controller, maka *method* tersebut akan memanggil *method* rekap() untuk mengambil data waktu yang berisi bulan dan tahun saat ini, lalu kembali ke *method* hitungRekap() dan memanggil *method* getAllOPD() pada *entity* user\_model untuk mendapatkan data OPD. Lalu *controller* statistik\_controller melakukan *looping* terhadap data OPD. Dalam *looping* tersebut, *controller* statistik\_controller memanggil *method* getLapAllOPD(id\_user, bulan, tahun) pada *entity* statistik\_model untuk mendapatkan data laporan tiap OPD dan data jumlah laporan tiap OPD. Lalu dilakukan *looping* terhadap data laporan tiap OPD, dimana *looping* tersebut memanggil *method* getLapDijawab(id\_user, kode\_laporan) untuk mendapatkan data jumlah laporan yang dijawab. Setelah *looping* selesai, *controller* statistik\_controller menampilkan rekapitulasi laporan pada bulan dan tahun saat ini melalui *boundary* rekap\_view.



**Gambar 5.4 Sequence Diagram Meneruskan Laporan**

Gambar 5.4 merupakan *sequence diagram* dari meneruskan laporan. Untuk melakukan aktivitas meneruskan laporan, aktor mengakses tombol teruskan pada *boundary* *moderasi\_view*, maka akan ditampilkan daftar OPD. Lalu aktor memilih OPD, lalu *boundary* *moderasi\_view* akan memanggil *method* *teruskan(kode\_laporan)* dan mengirimkan data OPD terpilih ke *controller* *admin\_controller*. Jika ada OPD yang dipilih, maka *controller* *admin\_controller* melakukan *looping* terhadap OPD terpilih, dalam *looping* tersebut, *controller* *admin\_controller* memanggil *method* *insertData(data\_insert)* pada *entity* *laporan\_model* untuk memasukkan data ke dalam tabel terusan. Di luar *looping* tersebut, *controller* *admin\_controller* memanggil *method* *updateData(data\_update, kode\_laporan)* pada *entity* *laporan\_model*. Lalu *controller* *admin\_controller* akan menampilkan pesan sukses melalui *boundary* *moderasi\_view*. Sedangkan jika tidak ada OPD yang dipilih, maka *admin\_controller* akan menampilkan pesan gagal melalui *boundary* *moderasi\_view*.

### 5.1.3 Pemodelan Class Diagram



Gambar 5.5 Class Diagram

Gambar 5.5 merupakan class diagram dari sistem aplikasi pelaporan masyarakat. Sistem ini menggunakan *framework* CodeIgniter, sehingga *class* *user\_controller*, *member\_controller*, *admin\_controller*, *opd\_controller*, *respon\_controller*, *kelolaOPD\_controller*, dan *statistik\_controller* melakukan *inheritance* terhadap *CI\_Controller*. Sedangkan *class* *laporan\_model*, *respon\_model*, *statistik\_model*, dan *user\_model* melakukan *inheritance* terhadap *CI\_Model*.

### 5.1.4 Perancangan Algoritme

Tabel 5.1 Pseudocode Melihat Statistik

1	Start method <code>hitungStatistik</code>
2	<code>waktu = memanggil method statistik</code>
3	<code>opd = data opd</code>
4	<code>jumlahLap = array()</code>
5	<code>nilai = array()</code>
6	<code>hasil=0</code>
7	<code>i=0</code>
8	<code>foreach opd</code>
9	<code>    laporan = ambil data laporan berdasarkan bulan dan tahun dari variabel waktu</code>
10	<code>    jumlahLap = ambil jumlah laporan pada bulan dan tahun dari variabel waktu</code>

11	tanggalMasuk = ambil tanggal masuk laporan pada bulan dan tahun dari variabel waktu
12	j = 0
13	nilai[i] = 0
14	foreach laporan
15	tj = ambil tanggal respon teratas
16	ttj = tj[tanggal_respon]
17	if tanggalMasuk != 0
18	masuk=tanggalMasuk[j][tanggal]
19	hasil = selisih masuk dan ttj
20	selisih = hasil dalam format hari
21	if selisih > 10, then 10, else selisih
22	endif
23	nilai[i] += 10 - selisih
24	increment j
25	else
26	nilai[i] = 0
27	end if
28	end foreach
29	if (j!=0) then nilai[i] /= j
30	increment i
31	end foreach
32	end

Tabel 5.1 merupakan *pseudocode* melihat statistik. *Pseudocode method* ini diawali dengan memanggil *method* statistik untuk mengambil data bulan dan tahun. Lalu terdapat perulangan terhadap data OPD untuk mengambil data laporan, jumlah laporan, dan tanggal laporan tiap OPD. Lalu dilakukan perulangan kembali terhadap data laporan untuk mendapatkan selisih hari dari tanggal laporan dan tanggal respon dan nilainya.

**Tabel 5.2 Pseudocode Melihat Rekapitulasi Laporan**

1	Start method hitungRekap
2	waktu = memanggil method rekap
3	opd = data opd
4	i = 0
5	laporanArray = array()
6	lapDijawabArray = array()
7	lapTidakDijawabArray = array()
8	nilaiAkhir = array()
9	j = 0
10	foreach opd
11	laporan = ambil jumlah laporan berdasarkan bulan dan tahun dari variabel waktu
12	laporanArray[i++] = laporan
13	lap = ambil data laporan berdasarkan bulan dan tahun dari variabel waktu
14	lapDijawabArray[j]=0
15	foreach lap
16	lapDijawab = ambil laporan yang dijawab
17	if lapDijawab > 0
18	lapDijawabArray[j] += 1
19	endif
20	end foreach
21	lapTidakDijawabArray[j] = laporan-lapDijawabArray[j]
22	nilai = lapDijawabArray[j]

23	if nilai<0, then 0, else nilai
24	endif
25	if laporan != 0
26	nilaiAkhir[j] = nilai/laporan*100
27	else
28	nilaiAkhir[j] = 0
29	end if
30	increment j
31	end foreach
32	end

Tabel 5.2 merupakan *pseudocode* melihat rekapitulasi laporan. *Pseudocode method* ini diawali dengan memanggil *method* rekap untuk mengambil data bulan dan tahun. Lalu terdapat perulangan terhadap data OPD untuk mengambil data laporan dan jumlah laporan tiap OPD. Lalu dilakukan perulangan kembali terhadap data laporan untuk mendapatkan jumlah laporan yang dijawab. Kemudian dilakukan operasi matematika untuk mendapatkan jumlah laporan yang tidak dijawab dan nilai.

**Tabel 5.3 Pseudocode Meneruskan Laporan**

1	Start method teruskan
2	opd = input(opd)
3	if opd not empty
4	foreach opd
5	data_insert = array(kode laporan,
6	selected opd)
7	masukkan data_insert ke database
8	end foreach
9	data_update = array(
10	"status_terusan" = "1"
11	)
12	update data_update ke database
13	Print alert 'Laporan Berhasil
14	Diteruskan'
15	else
16	Print alert 'Laporan Gagal Diteruskan'
17	end if
18	end

Tabel 5.3 merupakan *pseudocode* melihat meneruskan laporan. *Pseudocode method* ini diawali dengan mengambil data OPD yang dimasukkan admin. Lalu dilakukan perulangan terhadap data OPD yang dimasukkan untuk melakukan *insert* ke *database*. Selain melakukan fungsi *insert*, di dalam *method* ini juga dilakukan *update* status bahwa laporan tersebut telah diteruskan.

### 5.1.5 Perancangan Antarmuka

Perancangan antarmuka berisi perancangan berupa *wireframe* tiap halaman yang ada di dalam sistem aplikasi pelaporan masyarakat berbasis web.

#### 5.1.5.1 Halaman Beranda User

Gambar 5.6 merupakan perancangan antarmuka halaman beranda *user*. Penjelasan setiap objek yang ada di dalam perancangan antarmuka halaman beranda *user* dijelaskan pada Tabel 5.4.

Gambar 5.6 Perancangan Antarmuka Halaman Beranda *User*Tabel 5.4 Penjelasan Perancangan Antarmuka Halaman Beranda *User*

No	Nama Objek	Tipe	Keterangan
1	Logo	Gambar	Menampilkan logo aplikasi.
2	<i>Username</i>	<i>Text field</i>	<i>Field</i> untuk mengisi <i>username</i> .
3	<i>Password</i>	<i>Text field</i>	<i>Field</i> untuk mengisi <i>password</i> .
4	<i>Login</i>	Tombol	Tombol untuk <i>login</i> .
5	Daftar	Tombol	Tombol untuk mengakses halaman daftar.
6	Beranda	Navigasi	Navigasi untuk mengakses halaman beranda.
7	Cari Laporan	Navigasi	Navigasi untuk mengakses halaman cari laporan.
8	Statistik	Navigasi	Navigasi untuk mengakses halaman statistik.
9	Rekapitulasi	Navigasi	Navigasi untuk mengakses halaman rekapitulasi laporan.
10	Laporan	<i>Box</i>	Menampilkan laporan yang masuk ke dalam sistem.

#### 5.1.5.2 Halaman Daftar

Gambar 5.7 merupakan perancangan antarmuka halaman daftar. Penjelasan setiap objek yang ada di dalam perancangan antarmuka halaman daftar dijelaskan pada Tabel 5.5.



1		
Login	Daftarkan Diri Anda	
2	Nama	10
3	NIK	11
4	No Handphone	12
5		
MENU UTAMA		
6	Username	13
7	Password	14
8		15
9		

**Gambar 5.7 Perancangan Antarmuka Halaman Daftar**

**Tabel 5.5 Penjelasan Perancangan Antarmuka Halaman Daftar**

No	Nama Objek	Tipe	Keterangan
1	Logo	Gambar	Menampilkan logo aplikasi.
2	<i>Username</i>	<i>Text field</i>	<i>Field</i> untuk mengisi <i>username</i> .
3	<i>Password</i>	<i>Text field</i>	<i>Field</i> untuk mengisi <i>password</i> .
4	<i>Login</i>	Tombol	Tombol untuk <i>login</i> .
5	Daftar	Tombol	Tombol untuk mengakses halaman daftar.
6	Beranda	Navigasi	Navigasi untuk mengakses halaman beranda.
7	Cari Laporan	Navigasi	Navigasi untuk mengakses halaman cari laporan.
8	Statistik	Navigasi	Navigasi untuk mengakses halaman statistik.
9	Rekapitulasi	Navigasi	Navigasi untuk mengakses halaman rekapitulasi laporan.
10	Nama	<i>Text field</i>	<i>Field</i> untuk mengisi nama.
11	NIK	<i>Text field</i>	<i>Field</i> untuk mengisi NIK.
12	No. Handphone	<i>Text field</i>	<i>Field</i> untuk mengisi nomor <i>handphone</i> .
13	<i>Username</i>	<i>Text field</i>	<i>Field</i> untuk mengisi <i>username</i> .

14	Password	Text field	Field untuk mengisi password.
15	Daftar	Tombol	Tombol untuk mendaftar.

### 5.1.5.3 Halaman Cari Laporan

Gambar 5.8 merupakan perancangan antarmuka halaman cari laporan. Penjelasan setiap objek yang ada di dalam perancangan antarmuka halaman cari laporan dijelaskan pada Tabel 5.6.

Gambar 5.8 Perancangan Antarmuka Halaman Cari Laporan

Tabel 5.6 Penjelasan Perancangan Antarmuka Halaman Cari Laporan

No	Nama Objek	Tipe	Keterangan
1	Logo	Gambar	Menampilkan logo aplikasi.
2	Username	Text field	Field untuk mengisi username.
3	Password	Text field	Field untuk mengisi password.
4	Login	Tombol	Tombol untuk login.
5	Daftar	Tombol	Tombol untuk mengakses halaman daftar.
6	Beranda	Navigasi	Navigasi untuk mengakses halaman beranda.
7	Cari Laporan	Navigasi	Navigasi untuk mengakses halaman cari laporan.
8	Statistik	Navigasi	Navigasi untuk mengakses halaman statistik.

9	Rekapitulasi	Navigasi	Navigasi untuk mengakses halaman rekapitulasi laporan.
10	Kata Kunci	<i>Text field</i>	<i>Field</i> untuk mengisi kata kunci.
11	Subjek Laporan	<i>Text field</i>	<i>Field</i> untuk mengisi subjek laporan.
12	Nama Pengirim	<i>Text field</i>	<i>Field</i> untuk mengisi nama pengirim.
13	Cari	Tombol	Tombol untuk mencari laporan.

#### 5.1.5.4 Halaman Statistik

Gambar 5.9 merupakan perancangan antarmuka halaman statistik. Penjelasan setiap objek yang ada di dalam perancangan antarmuka halaman statistik dijelaskan pada Tabel 5.7.

**Gambar 5.9 Perancangan Antarmuka Halaman Statistik**

**Tabel 5.7 Penjelasan Perancangan Antarmuka Halaman Statistik**

No	Nama Objek	Tipe	Keterangan
1	Logo	Gambar	Menampilkan logo aplikasi.
2	<i>Username</i>	<i>Text field</i>	<i>Field</i> untuk mengisi <i>username</i> .
3	<i>Password</i>	<i>Text field</i>	<i>Field</i> untuk mengisi <i>password</i> .
4	<i>Login</i>	Tombol	Tombol untuk <i>login</i> .
5	Daftar	Tombol	Tombol untuk mengakses halaman daftar.

6	Beranda	Navigasi	Navigasi untuk mengakses halaman beranda.
7	Cari Laporan	Navigasi	Navigasi untuk mengakses halaman cari laporan.
8	Statistik	Navigasi	Navigasi untuk mengakses halaman statistik.
9	Rekapitulasi	Navigasi	Navigasi untuk mengakses halaman rekapitulasi laporan.
10	Pilih Bulan	<i>Dropdown</i>	<i>Dropdown</i> untuk memilih bulan.
11	Pilih Tahun	<i>Dropdown</i>	<i>Dropdown</i> untuk memilih tahun.
12	Pilih	Tombol	Tombol untuk menampilkan statistik berdasarkan bulan dan tahun terpilih.
13	Statistik	Diagram Batang	Menampilkan statistik berdasarkan bulan dan tahun terpilih.

#### 5.1.5.5 Halaman Rekapitulasi Laporan

Gambar 5.10 merupakan perancangan antarmuka halaman rekapitulasi laporan. Penjelasan setiap objek yang ada di dalam perancangan antarmuka halaman rekapitulasi laporan dijelaskan pada Tabel 5.8.

No	Nama OPD	Jumlah laporan yg diterima	Jumlah laporan yg dijawab	Jumlah laporan yg tidak dijawab	Nilai (%)
13	14	15	16	17	18

**Gambar 5.10 Perancangan Antarmuka Halaman Rekapitulasi Laporan**

Tabel 5.8 Penjelasan Perancangan Antarmuka Halaman Rekapitulasi Laporan

No	Nama Objek	Tipe	Keterangan
1	Logo	Gambar	Menampilkan logo aplikasi.
2	<i>Username</i>	<i>Text field</i>	<i>Field</i> untuk mengisi <i>username</i> .
3	<i>Password</i>	<i>Text field</i>	<i>Field</i> untuk mengisi <i>password</i> .
4	<i>Login</i>	Tombol	Tombol untuk <i>login</i> .
5	Daftar	Tombol	Tombol untuk mengakses halaman daftar.
6	Beranda	Navigasi	Navigasi untuk mengakses halaman beranda.
7	Cari Laporan	Navigasi	Navigasi untuk mengakses halaman cari laporan.
8	Statistik	Navigasi	Navigasi untuk mengakses halaman statistik.
9	Rekapitulasi	Navigasi	Navigasi untuk mengakses halaman rekapitulasi laporan.
10	Pilih Bulan	<i>Dropdown</i>	<i>Dropdown</i> untuk memilih bulan.
11	Pilih Tahun	<i>Dropdown</i>	<i>Dropdown</i> untuk memilih tahun.
12	Pilih	Tombol	Tombol untuk menampilkan statistik berdasarkan bulan dan tahun terpilih.
13	No.	<i>Text</i>	Menampilkan nomor data yang ditampilkan.
14	Nama OPD	<i>Text</i>	Menampilkan nama OPD.
15	Jumlah laporan yang diterima	<i>Text</i>	Menampilkan jumlah laporan yang diterima.
16	Jumlah laporan yang dijawab	<i>Text</i>	Menampilkan jumlah laporan yang dijawab.
17	Jumlah laporan yang tidak dijawab	<i>Text</i>	Menampilkan jumlah laporan yang tidak dijawab.
18	Nilai	<i>Text</i>	Menampilkan nilai.

#### 5.1.5.6 Halaman Formulir Pelaporan

Gambar 5.11 merupakan perancangan antarmuka halaman formulir pelaporan. Penjelasan setiap objek yang ada di dalam perancangan antarmuka halaman formulir pelaporan dijelaskan pada Tabel 5.9.

The screenshot shows a web application interface for reporting. On the left is a sidebar menu (1) with a 'MENU UTAMA' header and items 4 through 9. The main content area has a title bar (2) with a sub-title (3) 'Formulir Pelaporan'. Below this are four input fields: 'Subjek' (10), 'Isi' (11), 'Lokasi' (12), and 'Lampiran' (13). A button (14) is located at the bottom of the form area.

**Gambar 5.11 Perancangan Antarmuka Halaman Formulir Pelaporan**

**Tabel 5.9 Penjelasan Perancangan Antarmuka Halaman Formulir Pelaporan**

No	Nama Objek	Tipe	Keterangan
1	Logo	Gambar	Menampilkan logo aplikasi.
2	Logo aktor	Gambar	Menampilkan logo aktor.
3	Nama	Teks	Menampilkan nama aktor
4	Beranda	Navigasi	Navigasi untuk mengakses halaman beranda.
5	Cari Laporan	Navigasi	Navigasi untuk mengakses halaman cari laporan.
6	Formulir Pelaporan	Navigasi	Navigasi untuk mengakses halaman formulir pelaporan.
7	Laporan Saya	Navigasi	Navigasi untuk mengakses halaman riwayat laporan.
8	Statistik	Navigasi	Navigasi untuk mengakses halaman statistik.
9	Rekapitulasi	Navigasi	Navigasi untuk mengakses halaman rekapitulasi laporan.
10	Subjek	<i>Text field</i>	<i>Field</i> untuk mengisi subjek laporan.
11	Isi Laporan	<i>Text field</i>	<i>Field</i> untuk mengisi isi laporan.
12	Lokasi	<i>Text field</i>	<i>Field</i> untuk mengisi lokasi permasalahan.



13	File	Tombol	Tombol untuk membuka dialog <i>file</i> .
14	Kirim	Tombol	Tombol untuk mengirim laporan.

#### 5.1.5.7 Halaman Riwayat Laporan

Gambar 5.12 merupakan perancangan antarmuka halaman riwayat laporan. Penjelasan setiap objek yang ada di dalam perancangan antarmuka halaman riwayat laporan dijelaskan pada Tabel 5.10.

**Gambar 5.12 Perancangan Antarmuka Halaman Riwayat Laporan**

**Tabel 5.10 Penjelasan Perancangan Antarmuka Halaman Riwayat Laporan**

No	Nama Objek	Tipe	Keterangan
1	Logo	Gambar	Menampilkan logo aplikasi.
2	Logo aktor	Gambar	Menampilkan logo aktor.
3	Nama	Teks	Menampilkan nama aktor
4	Beranda	Navigasi	Navigasi untuk mengakses halaman beranda.
5	Cari Laporan	Navigasi	Navigasi untuk mengakses halaman cari laporan.
6	Formulir Pelaporan	Navigasi	Navigasi untuk mengakses halaman formulir pelaporan.
7	Laporan Saya	Navigasi	Navigasi untuk mengakses halaman riwayat laporan.
8	Statistik	Navigasi	Navigasi untuk mengakses halaman statistik.

9	Rekapitulasi	Navigasi	Navigasi untuk mengakses halaman rekapitulasi laporan.
10	Laporan	Box	Menampilkan riwayat laporan yang dikirimkan member.

#### 5.1.5.8 Halaman Rincian Laporan *Member*

Gambar 5.13 merupakan perancangan antarmuka halaman rincian laporan *member*. Penjelasan setiap objek yang ada di dalam perancangan antarmuka halaman rincian laporan *member* dijelaskan pada Tabel 5.11.

**Gambar 5.13 Perancangan Antarmuka Detail Laporan *Member***

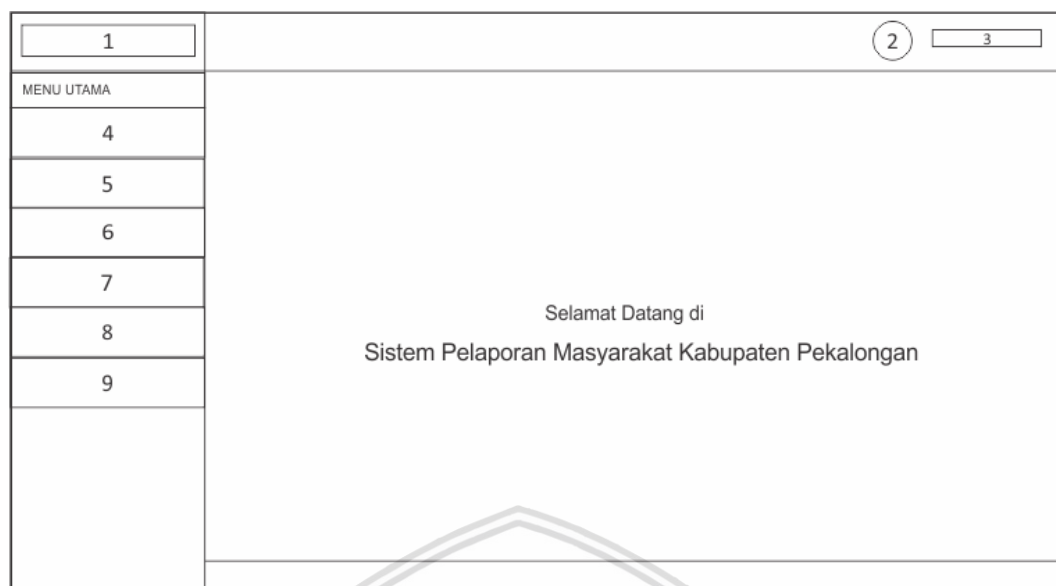
**Tabel 5.11 Penjelasan Perancangan Antarmuka Detail Laporan *Member***

No	Nama Objek	Tipe	Keterangan
1	Logo	Gambar	Menampilkan logo aplikasi.
2	Logo aktor	Gambar	Menampilkan logo aktor.
3	Nama	Teks	Menampilkan nama aktor
4	Beranda	Navigasi	Navigasi untuk mengakses halaman beranda.
5	Cari Laporan	Navigasi	Navigasi untuk mengakses halaman cari laporan.
6	Formulir Pelaporan	Navigasi	Navigasi untuk mengakses halaman formulir pelaporan.
7	Laporan Saya	Navigasi	Navigasi untuk mengakses halaman riwayat laporan.

8	Statistik	Navigasi	Navigasi untuk mengakses halaman statistik.
9	Rekapitulasi	Navigasi	Navigasi untuk mengakses halaman rekapitulasi laporan.
10	Subjek Laporan	Teks	Menampilkan riwayat subjek laporan.
11	Nama Pengirim	Teks	Menampilkan nama pengirim laporan.
12	Tanggal laporan	Teks	Menampilkan tanggal laporan.
13	Isi laporan	Teks	Menampilkan isi laporan.
14	Lampiran	Teks	Menampilkan nama <i>file</i> lampiran.
15	Buka lampiran	Tombol	Tombol untuk menampilkan lampiran.
16	Status laporan	Label	Menampilkan status keaktifan laporan.
17	Status Tahapan Laporan	List	Menampilkan status tahapan laporan.
18	Isi Respon	Text field	Field untuk mengisi respon balik.
19	File	Tombol	Tombol untuk membuka dialog <i>file</i> .
20	Kirim	Tombol	Tombol untuk mengirim respon balik.

#### 5.1.5.9 Halaman Beranda Admin

Gambar 5.14 merupakan perancangan antarmuka halaman beranda admin. Penjelasan setiap objek yang ada di dalam perancangan antarmuka halaman beranda admin dijelaskan pada Tabel 5.12.



**Gambar 5.14 Perancangan Antarmuka Halaman Beranda Admin**

**Tabel 5.12 Penjelasan Perancangan Antarmuka Halaman Beranda Admin**

No	Nama Objek	Tipe	Keterangan
1	Logo	Gambar	Menampilkan logo aplikasi.
2	Logo aktor	Gambar	Menampilkan logo aktor.
3	Nama	Teks	Menampilkan nama aktor
4	Beranda	Navigasi	Navigasi untuk mengakses halaman beranda.
5	Cari Laporan	Navigasi	Navigasi untuk mengakses halaman cari laporan.
6	Moderasi Laporan	Navigasi	Navigasi untuk mengakses halaman moderasi laporan.
7	Kelola Daftar OPD	Navigasi	Navigasi untuk mengakses halaman kelola daftar OPD.
8	Statistik	Navigasi	Navigasi untuk mengakses halaman statistik.
9	Rekapitulasi	Navigasi	Navigasi untuk mengakses halaman rekapitulasi laporan.

#### 5.1.5.10 Halaman Moderasi Laporan

Gambar 5.15 merupakan perancangan antarmuka halaman moderasi laporan. Penjelasan setiap objek yang ada di dalam perancangan antarmuka halaman moderasi laporan dijelaskan pada Tabel 5.13.

1	2	3																								
MENU UTAMA	Moderasi Laporan																									
4	Daftar Laporan Masuk																									
5	<table border="1"> <thead> <tr> <th>No</th> <th>Tanggal</th> <th>Subjek</th> <th>Isi</th> <th>Status</th> <th>Aksi</th> <th>Jawab</th> <th>Teruskan</th> </tr> </thead> <tbody> <tr> <td>10</td> <td>11</td> <td>12</td> <td>13</td> <td>14</td> <td>15</td> <td>16</td> <td>17</td> </tr> <tr> <td colspan="8"> </td> </tr> </tbody> </table>		No	Tanggal	Subjek	Isi	Status	Aksi	Jawab	Teruskan	10	11	12	13	14	15	16	17								
No	Tanggal	Subjek	Isi	Status	Aksi	Jawab	Teruskan																			
10	11	12	13	14	15	16	17																			
6																										
7																										
8																										
9																										

Gambar 5.15 Perancangan Antarmuka Halaman Moderasi Laporan

Tabel 5.13 Penjelasan Perancangan Antarmuka Halaman Moderasi Laporan

No	Nama Objek	Tipe	Keterangan
1	Logo	Gambar	Menampilkan logo aplikasi.
2	Logo aktor	Gambar	Menampilkan logo aktor.
3	Nama	Teks	Menampilkan nama aktor
4	Beranda	Navigasi	Navigasi untuk mengakses halaman beranda.
5	Cari Laporan	Navigasi	Navigasi untuk mengakses halaman cari laporan.
6	Moderasi Laporan	Navigasi	Navigasi untuk mengakses halaman moderasi laporan.
7	Kelola Daftar OPD	Navigasi	Navigasi untuk mengakses halaman kelola daftar OPD.
8	Formulir Pelaporan	Navigasi	Navigasi untuk mengakses halaman formulir pelaporan.
9	Laporan Saya	Navigasi	Navigasi untuk mengakses halaman riwayat laporan.
8	Statistik	Navigasi	Navigasi untuk mengakses halaman statistik.
9	Rekapitulasi	Navigasi	Navigasi untuk mengakses halaman rekapitulasi laporan.

10	No.	Teks	Menampilkan nomor data yang ditampilkan.
11	Tanggal	Teks	Menampilkan tanggal laporan.
12	Subjek Laporan	Teks	Menampilkan subjek laporan.
13	Isi	Teks	Menampilkan isi laporan.
14	Status	Label	Menampilkan status publikasi laporan.
15	Check	Tombol	Tombol untuk mengatur status laporan dipublikasikan atau tidak.
16	Jawab	Tombol	Tombol untuk mengakses halaman rincian laporan.
17	Teruskan	Tombol	Tombol untuk mengakses halaman <i>pop-up</i> terusan laporan.

#### 5.1.5.11 Halaman Meneruskan Laporan

Gambar 5.16 merupakan perancangan antarmuka halaman meneruskan laporan. Penjelasan setiap objek yang ada di dalam perancangan antarmuka halaman meneruskan laporan dijelaskan pada Tabel 5.14.

Gambar 5.16 Perancangan Antarmuka Meneruskan Laporan

Tabel 5.14 Penjelasan Perancangan Antarmuka Meneruskan Laporan

No	Nama Objek	Tipe	Keterangan
1	Checkbox	Checkbox	Checkbox untuk memilih OPD.
2	Nama OPD	Teks	Menampilkan nama OPD



3	Batal	Tombol	Tombol untuk membatalkan aksi.
4	Teruskan Laporan	Tombol	Tombol untuk meneruskan laporan.

#### 5.1.5.12 Halaman Kelola Daftar OPD

Gambar 5.17 merupakan perancangan antarmuka halaman kelola daftar OPD. Penjelasan setiap objek yang ada di dalam perancangan antarmuka halaman kelola daftar OPD dijelaskan pada Tabel 5.15.

**Gambar 5.17 Perancangan Antarmuka Halaman Kelola Daftar OPD**

**Tabel 5.15 Penjelasan Perancangan Antarmuka Halaman Kelola Daftar OPD**

No	Nama Objek	Tipe	Keterangan
1	Logo	Gambar	Menampilkan logo aplikasi.
2	Logo aktor	Gambar	Menampilkan logo aktor.
3	Nama	Teks	Menampilkan nama aktor
4	Beranda	Navigasi	Navigasi untuk mengakses halaman beranda.
5	Cari Laporan	Navigasi	Navigasi untuk mengakses halaman cari laporan.
6	Moderasi Laporan	Navigasi	Navigasi untuk mengakses halaman moderasi laporan.
7	Kelola Daftar OPD	Navigasi	Navigasi untuk mengakses halaman kelola daftar OPD.
8	Formulir Pelaporan	Navigasi	Navigasi untuk mengakses halaman formulir pelaporan.

9	Laporan Saya	Navigasi	Navigasi untuk mengakses halaman riwayat laporan.
8	Statistik	Navigasi	Navigasi untuk mengakses halaman statistik.
9	Rekapitulasi	Navigasi	Navigasi untuk mengakses halaman rekapitulasi laporan.
10	No.	Text	Menampilkan nomor data yang ditampilkan.
11	Username	Teks	Menampilkan <i>username</i> OPD.
12	Password	Teks	Menampilkan <i>password</i> OPD.
13	Nama OPD	Teks	Menampilkan nama OPD.
14	NIK	Teks	Menampilkan NIK OPD.
15	No. HP	Teks	Menampilkan nomor <i>handphone</i> OPD.
16	Edit	Tombol	Tombol untuk mengedit data OPD.
17	Hapus	Tombol	Tombol untuk menghapus data OPD.
18	Tambah	Tombol	Tombol untuk mengakses halaman <i>pop-up</i> tambah OPD.

#### 5.1.5.13 Halaman Edit OPD

Gambar 5.18 merupakan perancangan antarmuka halaman edit OPD. Penjelasan setiap objek yang ada di dalam perancangan antarmuka halaman edit OPD dijelaskan pada Tabel 5.16.

**Gambar 5.18 Perancangan Antarmuka Halaman Edit OPD**

**Tabel 5.16 Penjelasan Perancangan Antarmuka Halaman Edit OPD**

No	Nama Objek	Type	Keterangan
1	Nama OPD	<i>Text field</i>	<i>Field</i> untuk mengubah nama OPD.
2	NIK	<i>Text field</i>	<i>Field</i> untuk mengubah NIK OPD.
3	No. HP	<i>Text field</i>	<i>Field</i> untuk mengubah nomor <i>handphone</i> OPD.
4	<i>Username</i>	<i>Text field</i>	<i>Field</i> untuk mengubah <i>username</i> .
5	<i>Password</i>	<i>Text field</i>	<i>Field</i> untuk mengubah <i>password</i> .
6	Batal	Tombol	Tombol untuk membatalkan aksi.
7	Simpan	Tombol	Tombol untuk menyimpan data.

#### 5.1.5.14 Halaman Tambah OPD

Gambar 5.19 merupakan perancangan antarmuka halaman tambah OPD. Penjelasan setiap objek yang ada di dalam perancangan antarmuka halaman tambah OPD dijelaskan pada Tabel 5.17.

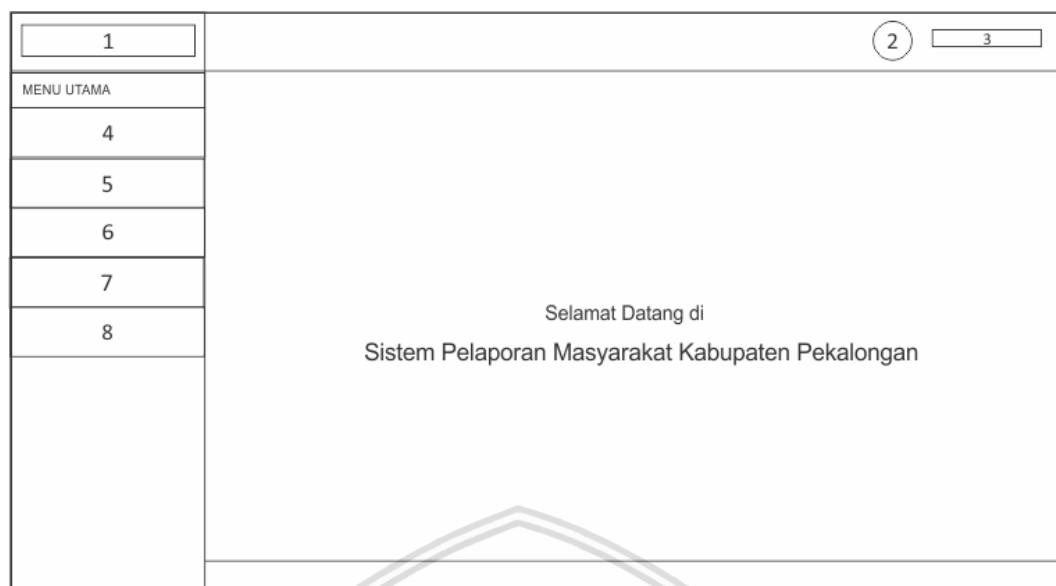
**Gambar 5.19 Perancangan Antarmuka Halaman Tambah OPD**

**Tabel 5.17 Penjelasan Perancangan Antarmuka Halaman Tambah OPD**

No	Nama Objek	Tipe	Keterangan
1	Nama OPD	<i>Text field</i>	<i>Field untuk mengisi nama OPD.</i>
2	NIK	<i>Text field</i>	<i>Field untuk mengisi NIK OPD.</i>
3	No. HP	<i>Text field</i>	<i>Field untuk mengisi nomor handphone OPD.</i>
4	<i>Username</i>	<i>Text field</i>	<i>Field untuk mengisi username.</i>
5	<i>Password</i>	<i>Text field</i>	<i>Field untuk mengisi password.</i>
6	Batal	Tombol	Tombol untuk membatalkan aksi.
7	Simpan	Tombol	Tombol untuk menyimpan data.

#### 5.1.5.15 Halaman Beranda OPD

Gambar 5.20 merupakan perancangan antarmuka halaman beranda OPD. Penjelasan setiap objek yang ada di dalam perancangan antarmuka halaman beranda OPD dijelaskan pada Tabel 5.18.



**Gambar 5.20 Perancangan Antarmuka Halaman Beranda OPD**

**Tabel 5.18 Penjelasan Perancangan Antarmuka Halaman Beranda OPD**

No	Nama Objek	Tipe	Keterangan
1	Logo	Gambar	Menampilkan logo aplikasi.
2	Logo aktor	Gambar	Menampilkan logo aktor.
3	Nama	Teks	Menampilkan nama aktor
4	Beranda	Navigasi	Navigasi untuk mengakses halaman beranda.
5	Cari Laporan	Navigasi	Navigasi untuk mengakses halaman cari laporan.
6	Laporan Masuk	Navigasi	Navigasi untuk mengakses halaman laporan masuk.
7	Statistik	Navigasi	Navigasi untuk mengakses halaman statistik.
8	Rekapitulasi	Navigasi	Navigasi untuk mengakses halaman rekapitulasi laporan.

#### 5.1.5.16 Halaman Laporan Masuk OPD

Gambar 5.21 merupakan perancangan antarmuka halaman laporan masuk untuk OPD. Penjelasan setiap objek yang ada di dalam perancangan antarmuka halaman beranda OPD dijelaskan pada Tabel 5.19.

1	2 3													
MENU UTAMA	Laporan Masuk													
4	Daftar Laporan Masuk													
5	<table border="1"> <tr> <th>No</th> <th>Tanggal</th> <th>Subjek</th> <th>Isi</th> <th>Jawab</th> </tr> <tr> <td>9</td> <td>10</td> <td>11</td> <td>12</td> <td>13</td> </tr> </table>				No	Tanggal	Subjek	Isi	Jawab	9	10	11	12	13
No	Tanggal	Subjek	Isi	Jawab										
9	10	11	12	13										
6														
7														
8														

**Gambar 5.21 Perancangan Antarmuka Halaman Laporan Masuk OPD**

**Tabel 5.19 Penjelasan Perancangan Antarmuka Halaman Laporan Masuk OPD**

No	Nama Objek	Tipe	Keterangan
1	Logo	Gambar	Menampilkan logo aplikasi.
2	Logo aktor	Gambar	Menampilkan logo aktor.
3	Nama	Teks	Menampilkan nama aktor
4	Beranda	Navigasi	Navigasi untuk mengakses halaman beranda.
5	Cari Laporan	Navigasi	Navigasi untuk mengakses halaman cari laporan.
6	Laporan Masuk	Navigasi	Navigasi untuk mengakses halaman laporan masuk.
7	Statistik	Navigasi	Navigasi untuk mengakses halaman statistik.
8	Rekapitulasi	Navigasi	Navigasi untuk mengakses halaman rekapitulasi laporan.
9	No.	Teks	Menampilkan nomor data yang ditampilkan.
10	Tanggal	Teks	Menampilkan tanggal laporan.
11	Subjek Laporan	Teks	Menampilkan subjek laporan.
12	Isi	Teks	Menampilkan isi laporan.
13	Jawab	Tombol	Tombol untuk mengakses halaman rincian laporan.



### 5.1.5.17 Halaman Rincian Laporan OPD

Gambar 5.22 merupakan perancangan antarmuka halaman beranda OPD. Penjelasan setiap objek yang ada di dalam perancangan antarmuka halaman beranda OPD dijelaskan pada Tabel 5.20.

The wireframe shows a web interface for the OPD Report Detail Page. It features a top navigation bar with a logo (1), a user profile section (2, 3), a main menu (4-8), a report title (9), a search bar (10), a report list table (11-21), and a status report box (16).

**Gambar 5.22 Perancangan Antarmuka Halaman Rincian Laporan OPD**

**Tabel 5.20 Penjelasan Perancangan Antarmuka Halaman Rincian Laporan OPD**

No	Nama Objek	Tipe	Keterangan
1	Logo	Gambar	Menampilkan logo aplikasi.
2	Logo aktor	Gambar	Menampilkan logo aktor.
3	Nama	Teks	Menampilkan nama aktor
4	Beranda	Navigasi	Navigasi untuk mengakses halaman beranda.
5	Cari Laporan	Navigasi	Navigasi untuk mengakses halaman cari laporan.
6	Laporan Masuk	Navigasi	Navigasi untuk mengakses halaman laporan masuk.
7	Statistik	Navigasi	Navigasi untuk mengakses halaman statistik.
8	Rekapitulasi	Navigasi	Navigasi untuk mengakses halaman rekapitulasi laporan.
9	Subjek Laporan	Teks	Menampilkan riwayat subjek laporan.

10	Nama Pengirim	Teks	Menampilkan nama pengirim laporan.
11	Tanggal laporan	Teks	Menampilkan tanggal laporan.
12	Isi laporan	Teks	Menampilkan isi laporan.
13	Lampiran	Teks	Menampilkan nama <i>file</i> lampiran.
14	Buka lampiran	Tombol	Tombol untuk menampilkan lampiran.
15	Status laporan	Label	Menampilkan status keaktifan laporan.
16	Status Tahapan Laporan	List	Menampilkan status tahapan laporan.
17	Isi Respon	Text field	Field untuk mengisi jawaban laporan.
18	File	Tombol	Tombol untuk membuka dialog <i>file</i> .
19	Estimasi Tanggal	Date picker	Memilih tanggal estimasi penyelesaian masalah.
20	Estimasi Waktu	Time picker	Memilih waktu estimasi penyelesaian masalah.
21	Kirim	Tombol	Tombol untuk menjawab laporan.

## 5.2 Implementasi

Pada bagian implementasi, penulis akan menjabarkan mengenai implementasi sistem aplikasi pelaporan masyarakat yang dilakukan sesuai dengan perancangan yang telah dibuat. Implementasi tersebut terdiri dari spesifikasi pengembangan sistem, implementasi kode program, dan implementasi antarmuka.

### 5.2.1 Spesifikasi Pengembangan Sistem

Spesifikasi pengembangan sistem ini terdiri dari spesifikasi perangkat keras dan spesifikasi perangkat lunak yang digunakan oleh penulis dalam mengembangkan sistem aplikasi pelaporan masyarakat berbasis web.

#### 5.2.1.1 Spesifikasi Perangkat Keras

Spesifikasi perangkat keras terdiri dari beberapa komponen seperti *processor*, *hard disk drive*, RAM, dan *system model* yang digunakan oleh penulis dalam mengembangkan sistem. Spesifikasi perangkat keras dapat dilihat pada Tabel 5.21.

Tabel 5.21 Spesifikasi Perangkat Keras

Nama Komponen	Spesifikasi
<i>Processor</i>	Intel Core i3-2365M
<i>Hard Disk Drive</i>	364GB
RAM	4.00GB
<i>System Model</i>	Dell Inspiron 3421

#### 5.2.1.2 Spesifikasi Perangkat Lunak

Spesifikasi perangkat lunak terdiri dari beberapa komponen seperti *operating system*, *programming language*, *text editor*, dan *server* yang digunakan oleh penulis dalam mengembangkan sistem. Spesifikasi perangkat lunak dapat dilihat pada Tabel 5.22.

Tabel 5.22 Spesifikasi Perangkat Lunak

Nama Komponen	Spesifikasi
<i>Operating System</i>	Windows 10 Pro
<i>Programming Language</i>	PHP
<i>Text Editor</i>	Notepad++
<i>Server</i>	XAMPP

#### 5.2.2 Implementasi Data

Implementasi data berisi *Data Definition Language* (DDL) dari basis data yang digunakan dalam pengembangan sistem aplikasi pelaporan masyarakat. Pada bagian ini, akan dijabarkan mengenai *query* yang digunakan untuk membuat tabel yang ada pada basis data “wadul”. DDL pada sistem aplikasi pelaporan masyarakat dapat dilihat pada Tabel 5.23.

Tabel 5.23 Implementasi Data

1	CREATE TABLE USER (
2	ID_USER INT(11) PRIMARY KEY AUTO_INCREMENT,
3	HAKAKSES VARCHAR(25),
4	USERNAME VARCHAR(250),
5	PASSWORD TEXT,
6	NAMA TEXT,
7	NIK VARCHAR(100),
8	NO_HP VARCHAR(100)
9	);
10	
11	CREATE TABLE LAPORAN (
12	KODE_LAPORAN INT(11) PRIMARY KEY AUTO_INCREMENT,
13	ID_USER INT(11) REFERENCES USER(ID_USER),
14	TANGGAL DATE,
15	SUBJEK VARCHAR(50),
16	TEKS TEXT,

```

17     LOKASI TEXT,
18     ATTCH VARCHAR(250),
19     STATUS ENUM('0','1','\",''),
20     STATUS_TERIMA ENUM('0','1','\",''),
21     STATUS_TERUSAN ENUM('0','1','\",''),
22     STATUS_RESPON ENUM('0','1','\",''),
23     TANGGAL_EXP DATE
24 );
25
26 CREATE TABLE RESPON (
27     KODE_RESPON INT(11) PRIMARY KEY AUTO_INCREMENT,
28     ID_USER INT (11) REFERENCES USER(ID_USER),
29     KODE_LAPORAN INT(11) REFERENCES LAPORAN(KODE_LAPORAN),
30     TANGGAL_RESPON DATE,
31     ISI_RESPON TEXT,
32     ATTCH VARCHAR(250),
33     ESTIMASI_TGL DATE,
34     ESTIMASI_WAKTU TIME
35 );
36
37 CREATE TABLE TERUSAN (
38     ID_TERUSAN INT(11) PRIMARY KEY AUTO_INCREMENT,
39     KODE_LAPORAN INT(11) REFERENCES LAPORAN(KODE_LAPORAN),
40     ID_USER INT (11) REFERENCES USER(ID_USER),
41 );

```

### 5.2.3 Implementasi Kode Program

Implementasi kode program berisi kode program dari sistem aplikasi pelaporan masyarakat berdasarkan *pseudocode* yang telah dibuat pada Bab 5.1.4.

#### 5.2.3.1 Implementasi Kode Program *Method* hitungStatistik

Nama fungsi: hitungStatistik()

Nama kelas: statistik\_controller

Source code:

**Tabel 5.24 Source Code Melihat Statistik**

```

1 public function hitungStatistik(){
2     $data['page'] = 'statistik_view';
3     $waktu = $this->statistik();
4     $opd = $this->user_model->getAlloPD();
5     $jumlahLap = array();
6     $nilai = array();
7     $hasil=0;
8     $i=0;
9     foreach ($opd as $o) {
10         $laporan = $this->statistik_model-
11             >getLapAlloPD($o['id_user'], $waktu['bulan'],
12                 $waktu['tahun'])->result_array();
13         $jumlahLap = $this->statistik_model-
14             >getLapAlloPD($o['id_user'], $waktu['bulan'],
15                 $waktu['tahun'])->num_rows();
16         $tanggalMasuk = $this->statistik_model-
17             >getTanggalLaporan($o['id_user'],

```

13	\$waktu['bulan'], \$waktu['tahun']);
14	\$j = 0;
15	\$nilai[\$i] = 0;
16	foreach (\$laporan as \$lap){
17	\$tj = \$this->statistik_model->
18	getTanggalJawabTeratas(\$o['id_user'],\$lap['kode_lapo
19	ran']);
20	\$ttj = \$tj['tanggal_respon'];
21	if(\$tanggalMasuk != 0){
22	\$masuk=\$tanggalMasuk[\$j]['tanggal'];
23	\$hasil=date_diff(date_create(\$masuk),
24	date_create(\$ttj));
25	\$selisih = (int)(\$hasil->format("%d"));
26	\$selisih = \$selisih>10?10:\$selisih;
27	\$nilai[\$i] += 10 - \$selisih;
28	\$j++;
29	} else {
30	\$nilai[\$i] = 0;
31	break;
32	}
33	if (\$j!=0) \$nilai[\$i] /= \$j;
34	\$i++;
35	}
36	\$data['tahun'] = \$this->statistik_model->getTahun();
37	\$data['bulanhuruf'] = \$waktu['bulanhuruf'];
38	\$data['tahunhuruf'] = \$waktu['tahun'];
39	\$data['opd'] = \$this->user_model->getAlloPD();
	\$data['nilai'] = \$nilai;
	\$this->load->view('master_view', \$data);
	}

Penjelasan:

1	Deklarasi fungsi hitungStatistik
2	Inisialisasi array data untuk menampung nilai page
3	Inisialisasi variabel waktu untuk menampung array dari pemanggilan method statistik()
4	Inisialisasi variabel opd untuk menampung data opd
5	Inisialisasi array jumlahLap
6	Inisialisasi array nilai
7	Inisialisasi variabel hasil
8	Inisialisasi variabel i
9-32	Perulangan terhadap data opd untuk mengambil data laporan, jumlah laporan, dan tanggal laporan yang diterima tiap OPD. Di dalam perulangan tersebut terdapat perulangan terhadap data laporan untuk mendapatkan tanggal respon teratas, selisih hari dari tanggal laporan dan tanggal statistic penghitungan nilainya. Nilai tersebut dihitung rata-ratanya untuk mendapatkan nilai akhir.
33	Inisialisasi array data untuk menampung tahun
34	Inisialisasi array data untuk menampung bulanhuruf
35	Inisialisasi array data untuk menampung tahunhuruf
36	Inisialisasi array data untuk menampung opd

37	Inisialisasi array data untuk menampung nilai
38	Load view “master_view” dan mengirimkan array data

### 5.2.3.2 Implementasi Kode Program *Method* hitungRekap

Nama fungsi: `hitungRekap()`

Nama kelas: `statistik_controller`

Source code:

**Tabel 5.25 Source Code Melihat Rekapitulasi Laporan**

1	<code>public function hitungRekap() {</code>
2	<code>    \$data['page'] = 'rekap_view';</code>
3	<code>    \$waktu = \$this-&gt;rekap();</code>
4	<code>    \$opd = \$this-&gt;user_model-&gt;getAllOPD();</code>
5	<code>    \$i = 0;</code>
6	<code>    \$laporanArray = array();</code>
7	<code>    \$lapDijawabArray = array();</code>
8	<code>    \$lapTidakDijawabArray = array();</code>
9	<code>    \$nilaiAkhir = array();</code>
10	<code>    \$j = 0;</code>
11	<code>    foreach(\$opd as \$o){</code>
12	<code>        \$laporan = \$this-&gt;statistik_model-&gt;</code>
	<code>        &gt;getLapAllOPD(\$o['id_user'], \$waktu['bulan'],</code>
	<code>        \$waktu['tahun'])-&gt;num_rows();</code>
13	<code>        \$laporanArray[\$i++] = \$laporan;</code>
14	<code>        \$lap = \$this-&gt;statistik_model-&gt;</code>
	<code>        &gt;getLapAllOPD(\$o['id_user'], \$waktu['bulan'],</code>
	<code>        \$waktu['tahun'])-&gt;result_array();</code>
15	<code>        \$lapDijawabArray[\$j]=0;</code>
16	<code>        foreach(\$lap as \$l){</code>
17	<code>            \$lapDijawab = \$this-&gt;statistik_model-&gt;</code>
	<code>            &gt;getLapDijawab(\$o['id_user'],</code>
	<code>            \$l['kode_laporan']);</code>
18	<code>            if(\$lapDijawab &gt; 0){</code>
19	<code>                \$lapDijawabArray[\$j] += 1;</code>
20	<code>            }</code>
21	<code>        }</code>
22	<code>        \$lapTidakDijawabArray[\$j] = \$laporan -</code>
	<code>        \$lapDijawabArray[\$j];</code>
23	<code>        \$nilai = \$lapDijawabArray[\$j];</code>
24	<code>        \$nilai = \$nilai&lt;0?0:\$nilai;</code>
25	<code>        if(\$laporan != 0){</code>
26	<code>            \$nilaiAkhir[\$j] = \$nilai/\$laporan*100;</code>
27	<code>        } else{</code>
28	<code>            \$nilaiAkhir[\$j] = 0;</code>
29	<code>        }</code>
30	<code>    \$j++;</code>
31	<code>    }</code>
32	<code>    \$data['opd'] = \$this-&gt;user_model-&gt;getAllOPD();</code>
33	<code>    \$data['laporan'] = \$laporanArray;</code>
34	<code>    \$data['lapDijawab'] = \$lapDijawabArray;</code>
35	<code>    \$data['lapTidakDijawab'] = \$lapTidakDijawabArray;</code>
36	<code>    \$data['nilai'] = \$nilaiAkhir;</code>
37	<code>    \$data['tahun'] = \$this-&gt;statistik_model-&gt;getTahun();</code>



38	\$data['bulanhuruf'] = \$waktu['bulanhuruf'];
39	\$data['tahunhuruf'] = \$waktu['tahun'];
40	\$this->load->view('master_view', \$data);
41	}

Penjelasan:

1	Deklarasi fungsi hitungRekap
2	Inisialisasi array data untuk menampung nilai page
3	Inisialisasi variabel waktu untuk menampung array dari pemanggilan method rekap()
4	Inisialisasi variabel opd untuk menampung data opd
5	Inisialisasi variabel i
6	Inisialisasi array laporanArray
7	Inisialisasi array lapDijawabArray
8	Inisialisasi array lapTidakDijawabArray
9	Inisialisasi array nilaiAkhir
10	Inisialisasi variabel j
11-31	Perulangan terhadap data opd untuk mengambil data laporan dan jumlah laporan diterima tiap OPD. Di dalam perulangan tersebut terdapat perulangan terhadap data laporan untuk mendapatkan jumlah laporan yang dijawab. Setelah itu dilakukan penghitungan untuk mendapatkan jumlah laporan yang tidak dijawab dan nilai yang didapat tiap OPD.
32	Inisialisasi array data untuk menampung opd
33	Inisialisasi array data untuk menampung laporanArray
34	Inisialisasi array data untuk menampung lapDijawabArray
35	Inisialisasi array data untuk menampung lapTidakDijawabArray
36	Inisialisasi array data untuk menampung nilaiAkhir
37	Inisialisasi array data untuk menampung tahun
38	Inisialisasi array data untuk menampung bulanhuruf
39	Inisialisasi array data untuk menampung tahunhuruf
40	Load view "master_view" dan mengirimkan array data

### 5.2.3.3 Implementasi Kode Program *Method* teruskan

Nama fungsi: teruskan()

Nama kelas: admin\_controller

Source code:

**Tabel 5.26 Source Code Meneruskan Laporan**

1	public function teruskan(\$kode_laporan){
2	\$data['kode_laporan'] = \$kode_laporan;
3	\$opd = \$this->input->post('opd');
4	if(!empty(\$opd)){
5	foreach(\$opd as \$selected){
6	\$data_insert = array(
7	"kode_laporan" =>\$kode_laporan,

8	<code>"id_user" =&gt; \$selected,</code>
9	<code>);</code>
10	<code>\$this-&gt;laporan_model-&gt;InsertData(\$data_insert);</code>
11	<code>}</code>
12	<code>\$data_update = array(</code>
13	<code>    "status_terusan" =&gt; "1"</code>
14	<code>);</code>
15	<code>\$this-&gt;laporan_model-&gt;UpdateData(\$data_update, \$data);</code>
16	<code>\$this-&gt;session-&gt;set_flashdata("pesan",</code> <code>"&lt;script&gt;alert('Laporan Berhasil</code> <code>Diteruskan');&lt;/script&gt;");</code>
17	<code>redirect(base_url(). 'moderasi');</code>
18	<code>} else{</code>
19	<code>    \$this-&gt;session-&gt;set_flashdata("pesan",</code> <code>    "&lt;script&gt;alert('Laporan Gagal</code> <code>    Diteruskan');&lt;/script&gt;");</code>
20	<code>    redirect(base_url(). 'moderasi');</code>
21	<code>}</code>
22	<code>}</code>

Penjelasan:

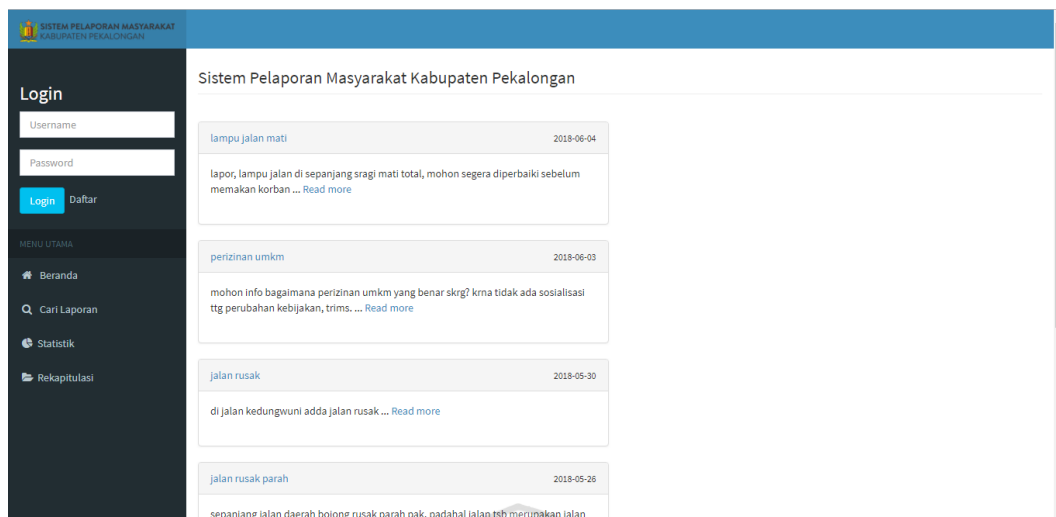
1	Deklarasi fungsi teruskan berparameter kode_laporan
2	Inisialisasi array data yang menampung kode laporan
3	Inisialisasi variabel opd yang menampung nilai input opd
4-21	Pengecekan kondisi jika variabel opd tidak kosong, maka melakukan perulangan terhadap nilai opd yang berisi fungsi <i>insert</i> opd yang dipilih ke dalam <i>database</i> . Setelah itu dilakukan fungsi <i>update</i> atribut <i>status_terusan</i> pada tabel laporan di <i>database</i> . Lalu, menampilkan pesan sukses atau tidaknya laporan diteruskan dan melakukan <i>redirect</i> ke halaman moderasi.

## 5.2.4 Implementasi Antarmuka

Implementasi antarmuka sistem merupakan penerapan dari perancangan antarmuka yang telah dibuat pada Bab 5.1.5.

### 5.2.4.1 Halaman Beranda User

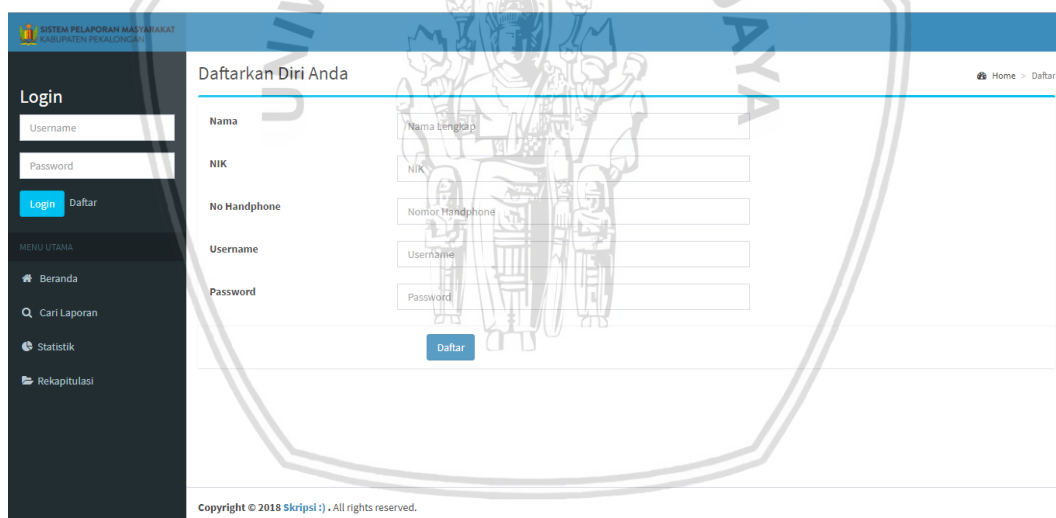
Halaman beranda *user* pada sistem aplikasi pelaporan masyarakat merupakan halaman yang pertama kali ditampilkan ketika mengakses sistem. Implementasi antarmuka halaman beranda *user* ditunjukkan pada Gambar 5.23.



**Gambar 5.23 Implementasi Antarmuka Halaman Beranda User**

#### 5.2.4.2 Halaman Daftar

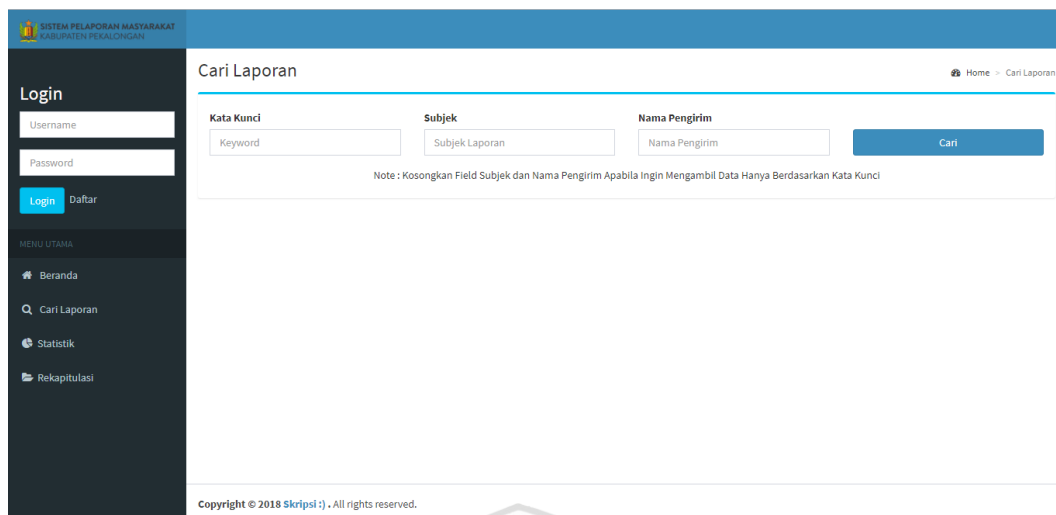
Halaman daftar pada sistem aplikasi pelaporan masyarakat merupakan halaman yang digunakan *user* untuk mendaftar sebagai *member*. Implementasi antarmuka halaman daftar ditunjukkan pada Gambar 5.24.



**Gambar 5.24 Implementasi Antarmuka Halaman Daftar**

#### 5.2.4.3 Halaman Cari Laporan

Halaman cari laporan pada sistem aplikasi pelaporan masyarakat merupakan halaman yang digunakan untuk mencari laporan berdasarkan kata kunci, subjek, dan/atau nama pengirim. Implementasi antarmuka halaman cari laporan ditunjukkan pada Gambar 5.25.



**SISTEM PELAPORAN MASYARAKAT KABUPATEN PEKALONGAN**

**Cari Laporan** Home > Cari Laporan

**Login**

Username

Password

[Login](#) [Daftar](#)

**MENU UTAMA**

- [Beranda](#)
- [Cari Laporan](#)
- [Statistik](#)
- [Rekapitulasi](#)

**Kata Kunci**

**Subjek**

**Nama Pengirim**

[Cari](#)

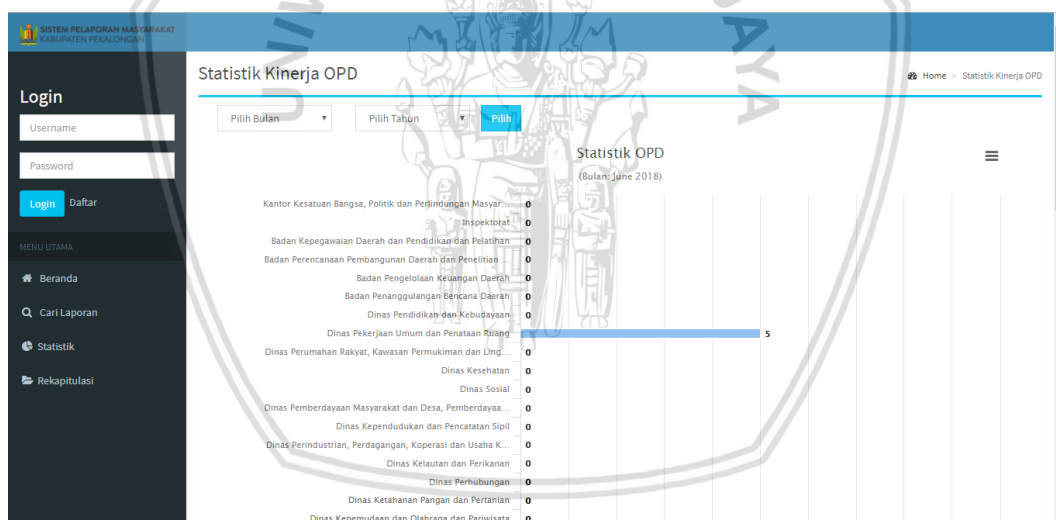
Note : Kosongkan Field Subjek dan Nama Pengirim Apabila Ingin Mengambil Data Hanya Berdasarkan Kata Kunci

Copyright © 2018 Skripsi j. All rights reserved.

**Gambar 5.25 Implementasi Antarmuka Halaman Cari Laporan**

#### 5.2.4.4 Halaman Statistik

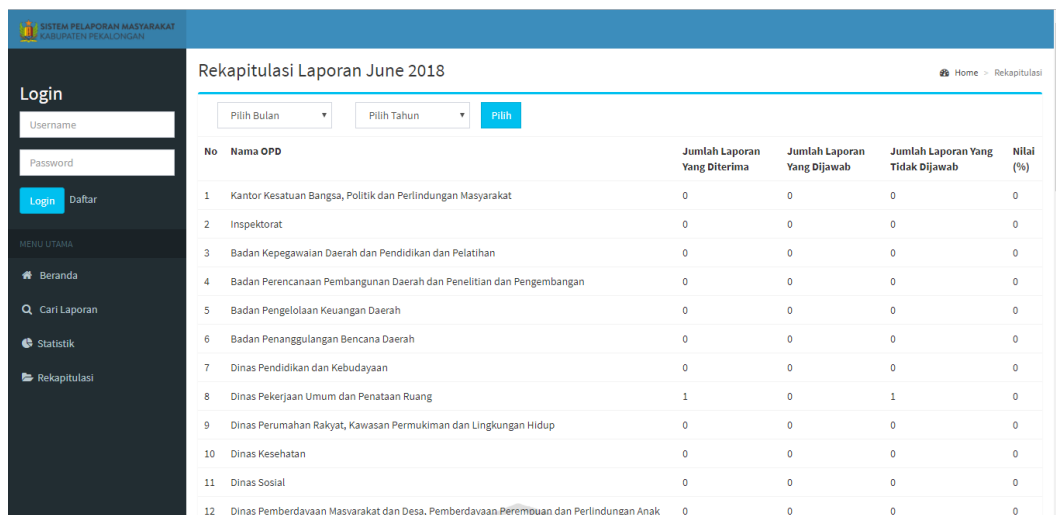
Halaman statistik pada sistem aplikasi pelaporan masyarakat merupakan halaman yang digunakan untuk melihat statistik kinerja OPD yang ditampilkan tiap bulan. Implementasi antarmuka halaman statistik ditunjukkan pada Gambar 5.26.



**Gambar 5.26 Implementasi Antarmuka Halaman Statistik**

#### 5.2.4.5 Halaman Rekapitulasi Laporan

Halaman rekapitulasi laporan pada sistem aplikasi pelaporan masyarakat merupakan halaman yang digunakan untuk melihat rekapitulasi laporan yang ditampilkan tiap bulan. Implementasi antarmuka halaman statistik ditunjukkan pada Gambar 5.27.

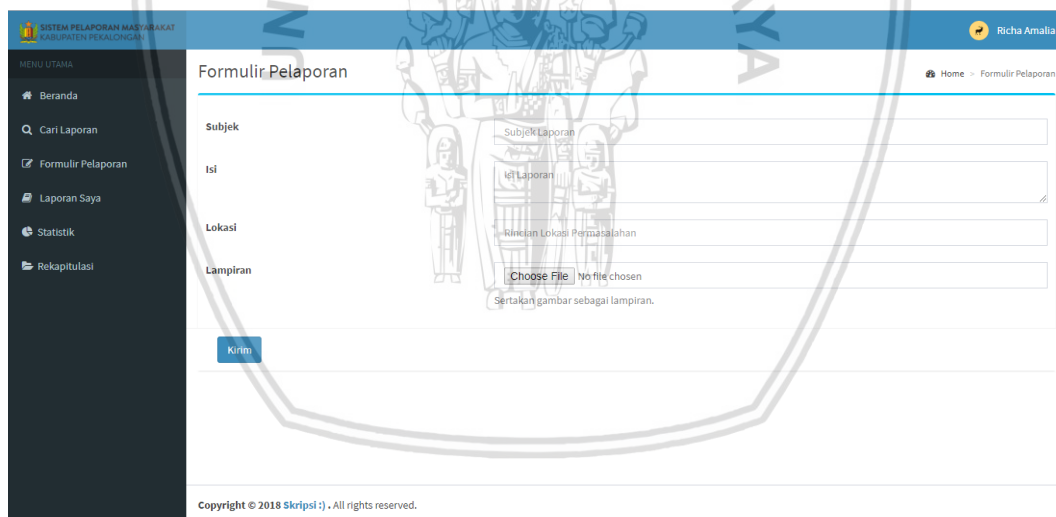


No	Nama OPD	Jumlah Laporan Yang Diterima	Jumlah Laporan Yang Dijawab	Jumlah Laporan Yang Tidak Dijawab	Nilai (%)
1	Kantor Kesatuan Bangsa, Politik dan Perlindungan Masyarakat	0	0	0	0
2	Inspektorat	0	0	0	0
3	Badan Kepegawaian Daerah dan Pendidikan dan Pelatihan	0	0	0	0
4	Badan Perencanaan Pembangunan Daerah dan Penelitian dan Pengembangan	0	0	0	0
5	Badan Pengelolaan Keuangan Daerah	0	0	0	0
6	Badan Penanggulangan Bencana Daerah	0	0	0	0
7	Dinas Pendidikan dan Kebudayaan	0	0	0	0
8	Dinas Pekerjaan Umum dan Penataan Ruang	1	0	1	0
9	Dinas Perumahan Rakyat, Kawasan Permukiman dan Lingkungan Hidup	0	0	0	0
10	Dinas Kesehatan	0	0	0	0
11	Dinas Sosial	0	0	0	0
12	Dinas Pemberdayaan Masyarakat dan Desa, Pemberdayaan Perempuan dan Perlindungan Anak	0	0	0	0

**Gambar 5.27 Implementasi Antarmuka Halaman Rekapitulasi Laporan**

#### 5.2.4.6 Halaman Formulir Pelaporan

Halaman formulir pelaporan pada sistem aplikasi pelaporan masyarakat merupakan halaman yang digunakan *member* untuk melaporkan permasalahan. Implementasi antarmuka halaman formulir pelaporan ditunjukkan pada Gambar 5.28.



Formulir Pelaporan

Subjek:

Isi:

Lokasi:

Lampiran:   No file chosen

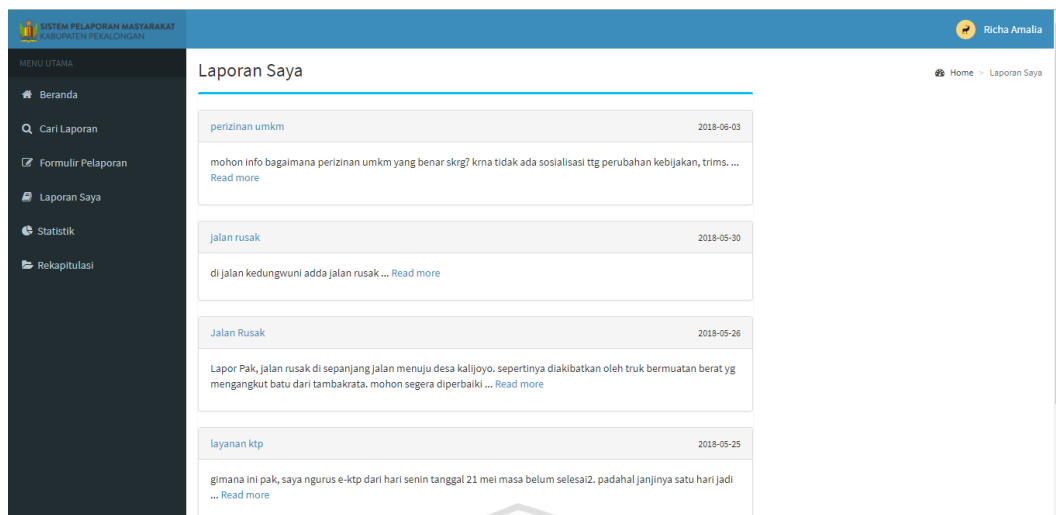
Sertakan gambar sebagai lampiran.

Copyright © 2018 skripsi :). All rights reserved.

**Gambar 5.28 Implementasi Antarmuka Halaman Formulir Pelaporan**

#### 5.2.4.7 Halaman Riwayat Laporan

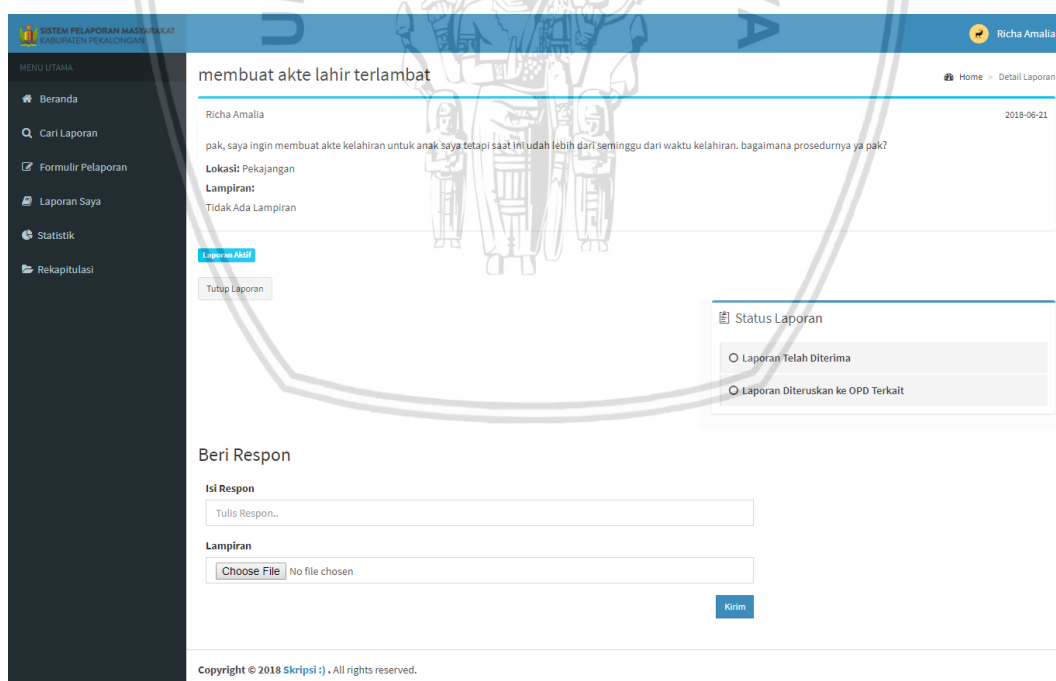
Halaman riwayat laporan pada sistem aplikasi pelaporan masyarakat merupakan halaman yang digunakan *member* untuk melihat riwayat laporan permasalahan yang pernah dikirimkan. Implementasi antarmuka halaman riwayat laporan ditunjukkan pada Gambar 5.29.



Gambar 5.29 Implementasi Antarmuka Halaman Riwayat Laporan

#### 5.2.4.8 Halaman Rincian Laporan *Member*

Halaman rincian laporan *member* pada sistem aplikasi pelaporan masyarakat merupakan halaman yang digunakan *member* untuk melihat rincian laporan, status tahapan laporan, status keaktifan laporan, menutup laporan, dan memberi respon balik. Implementasi antarmuka halaman rincian laporan *member* ditunjukkan pada Gambar 5.30.



Gambar 5.30 Implementasi Antarmuka Halaman Rincian Laporan *Member*

#### 5.2.4.9 Halaman Beranda Admin

Halaman beranda admin pada sistem aplikasi pelaporan masyarakat merupakan halaman yang ditampilkan setelah *user* melakukan login sebagai



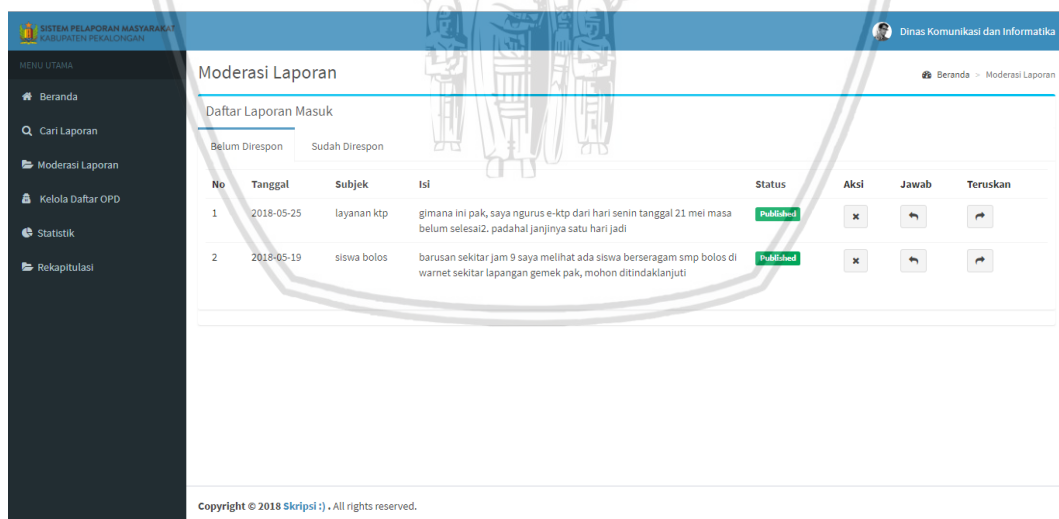
admin. Implementasi antarmuka halaman beranda admin ditunjukkan pada Gambar 5.31.



**Gambar 5.31 Implementasi Antarmuka Halaman Beranda Admin**

#### 5.2.4.10 Halaman Moderasi Laporan

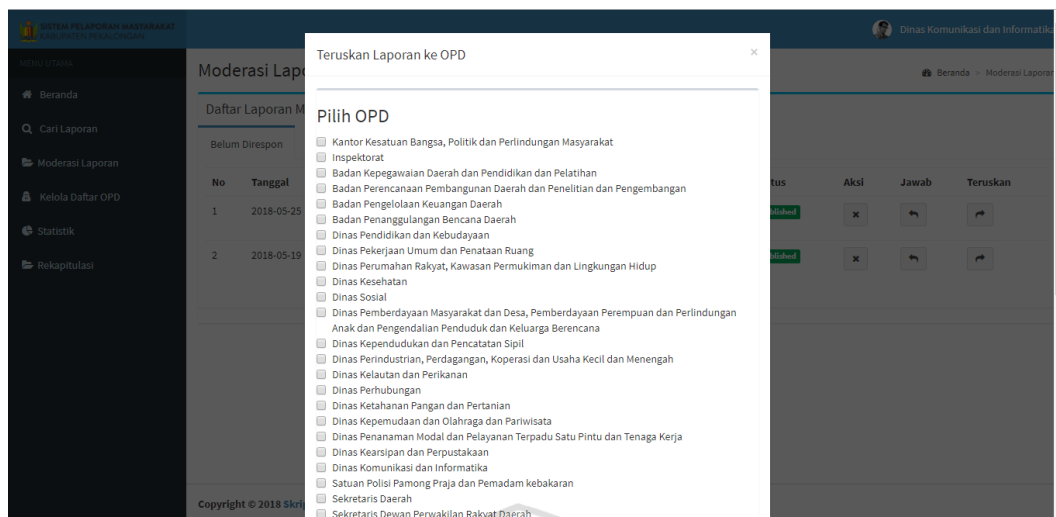
Halaman moderasi laporan pada sistem aplikasi pelaporan masyarakat merupakan halaman yang digunakan admin untuk melihat semua laporan yang masuk, melakukan moderasi laporan, menjawab laporan, dan meneruskan laporan ke OPD. Implementasi antarmuka halaman moderasi laporan ditunjukkan pada Gambar 5.32.



**Gambar 5.32 Implementasi Antarmuka Halaman Moderasi Laporan**

#### 5.2.4.11 Halaman Meneruskan Laporan

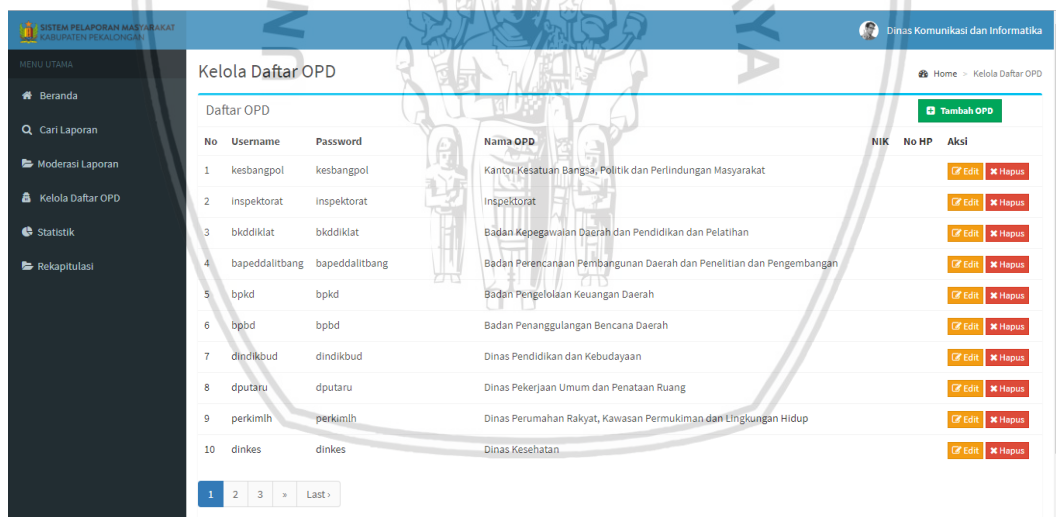
Halaman statistik pada sistem aplikasi pelaporan masyarakat merupakan halaman *pop-up* yang digunakan admin untuk memilih OPD yang terkait dengan permasalahan. Implementasi antarmuka halaman meneruskan laporan ditunjukkan pada Gambar 5.33.



**Gambar 5.33 Implementasi Antarmuka Halaman Meneruskan Laporan**

#### 5.2.4.12 Halaman Kelola Daftar OPD

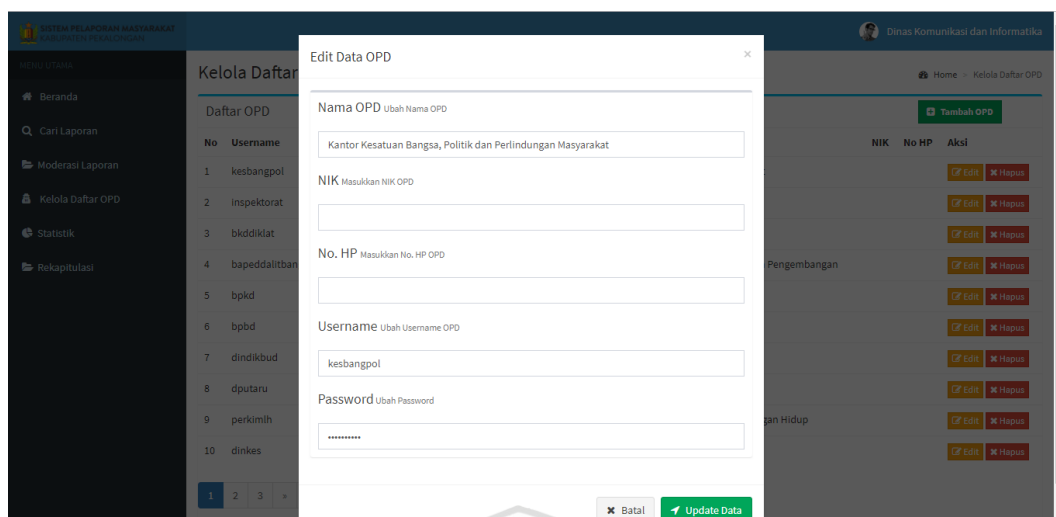
Halaman kelola daftar OPD pada sistem aplikasi pelaporan masyarakat merupakan halaman yang digunakan admin untuk melihat daftar OPD. Implementasi antarmuka halaman kelola daftar OPD ditunjukkan pada Gambar 5.34.



**Gambar 5.34 Implementasi Antarmuka Halaman Kelola Daftar OPD**

#### 5.2.4.13 Halaman Edit OPD

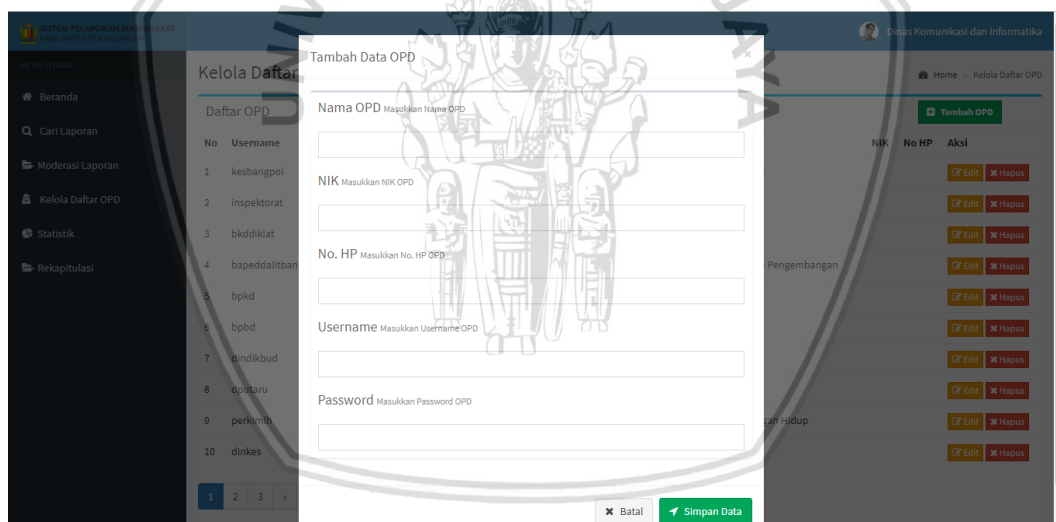
Halaman edit OPD pada sistem aplikasi pelaporan masyarakat merupakan halaman *pop-up* yang digunakan admin untuk mengubah data OPD. Implementasi antarmuka halaman edit OPD ditunjukkan pada Gambar 5.35.



Gambar 5.35 Implementasi Antarmuka Halaman Edit OPD

#### 5.2.4.14 Halaman Tambah OPD

Halaman tambah OPD pada sistem aplikasi pelaporan masyarakat merupakan halaman *pop-up* yang digunakan admin untuk menambah OPD. Implementasi antarmuka halaman tambah OPD ditunjukkan pada Gambar 5.36.



Gambar 5.36 Implementasi Antarmuka Halaman Tambah OPD

#### 5.2.4.15 Halaman Beranda OPD

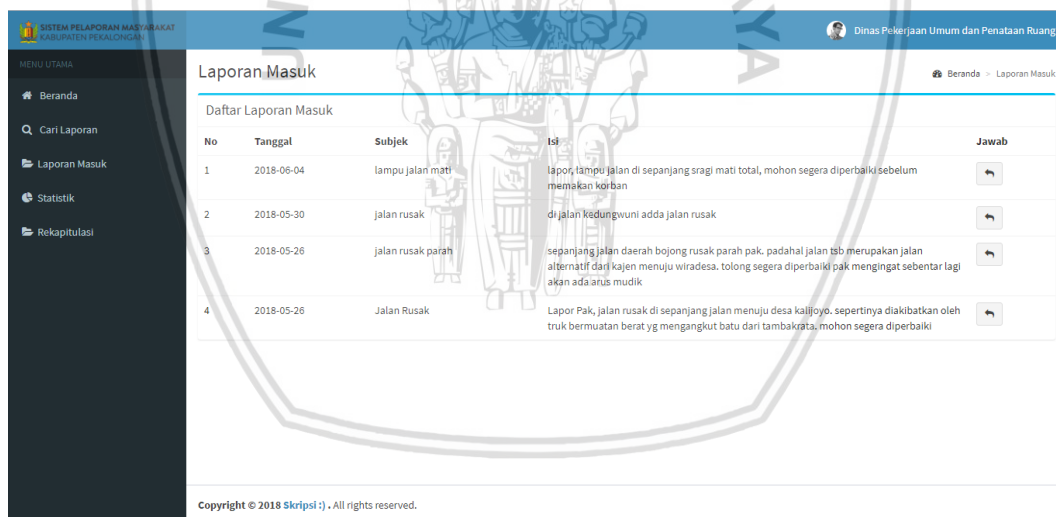
Halaman beranda OPD pada sistem aplikasi pelaporan masyarakat merupakan halaman yang ditampilkan setelah *user* melakukan login sebagai OPD. Implementasi antarmuka halaman beranda OPD ditunjukkan pada Gambar 5.37.



**Gambar 5.37 Implementasi Antarmuka Halaman Beranda OPD**

#### 5.2.4.16 Halaman Laporan Masuk


Halaman laporan masuk pada sistem aplikasi pelaporan masyarakat merupakan halaman yang digunakan OPD untuk melihat laporan yang ditujukan ke OPD tersebut dan menjawab laporan. Implementasi antarmuka halaman laporan masuk ditunjukkan pada Gambar 5.38.




**Gambar 5.38 Implementasi Antarmuka Halaman Laporan Masuk**

#### 5.2.4.17 Halaman Rincian Laporan OPD

Halaman rincian laporan OPD pada sistem aplikasi pelaporan masyarakat merupakan halaman yang digunakan OPD untuk melihat rincian laporan, status tahapan laporan, status keaktifan laporan, dan menjawab laporan. Implementasi antarmuka halaman rincian laporan OPD ditunjukkan pada Gambar 5.39.


**SISTEM LAPORAN MASYARAKAT**  
 KABUPATEN PECALANGAN


 Dinas Kependudukan dan Pencatatan Sipil

**MENU UTAMA**

- Beranda
- Cari Laporan
- Laporan Masuk
- Statistik
- Rekapitulasi

### membuat akte lahir terlambat

[Home](#) > [Detail Laporan](#)

Richa Amalia

2018-06-21

pak, saya ingin membuat akte kelahiran untuk anak saya tetapi saat ini udah lebih dari seminggu dari waktu kelahiran. bagaimana prosedurnya ya pak?

**Lokasi:** Pekajangan

**Lampiran:**  
Tidak Ada Lampiran

Laporan Aktif

Status Laporan

☐ Laporan Telah Diterima
 ☐ Laporan Diteruskan ke OPD Terkait

Beri Respon

Isi Respon

Tulis Respon...

Lampiran

Choose File

No file chosen

Estimasi Tanggal:

Estimasi Waktu:

12:30 AM

Kirim

Copyright © 2018 Skripsi31. All rights reserved.

Gambar 5.39 Implementasi Antarmuka Halaman Laporan Masuk

## BAB 6 PENGUJIAN

Pengujian merupakan tahap yang dilakukan setelah melakukan implementasi. Pada tahap ini, dilakukan pemeriksaan terhadap hasil implementasi untuk mengetahui apakah sistem yang diimplementasikan sudah sesuai dengan analisis kebutuhan dan perancangan. Pengujian yang dilakukan berupa pengujian unit, pengujian integrasi, pengujian validasi, dan pengujian *compatibility*.

### 6.1 Pengujian Unit

Pengujian unit merupakan pengujian yang dilakukan dengan menguji setiap unit dari sistem seperti komponen, kelas, atau objek. Pengujian unit ini merupakan pengujian dengan metode *white-box* dengan menggunakan jenis *basis path testing*.

#### 6.1.1 Pengujian Unit *Method* hitungStatistik

##### 1. Pseudocode

**Tabel 6.1 Pseudocode Pengujian *Method* hitungStatistik**

```

Start method hitungStatistik (1)
waktu = memanggil method statistik
opd = data opd
jumlahLap = array()
nilai = array() } (2)
hasil=0
i=0
foreach opd (3)
    laporan = ambil data laporan berdasarkan bulan dan
    tahun dari variabel waktu
    jumlahLap = ambil jumlah laporan pada bulan dan tahun
    dari variabel waktu
    tanggalMasuk = ambil tanggal masuk laporan pada bulan
    dan tahun dari variabel waktu
    j = 0
    nilai[i] = 0
    foreach laporan (5)
        tj = ambil tanggal respon teratas
        ttj = tj[tanggal_respon]
        if tanggalMasuk != 0 (7)
            masuk=tanggalMasuk[j][tanggal]
            hasil = selisih masuk dan ttj
            selisih = hasil dalam format hari } (8)
            if selisih>10, (9) then 10 (10), else selisih(11)
            endif (12)
            nilai[i] += 10 - selisih } (13)
            increment j
        else
            nilai[i] = 0 } (14)
        end if (15)
    end foreach (16)
    if (j!=0) (17) then nilai[i] /= j (18)

```



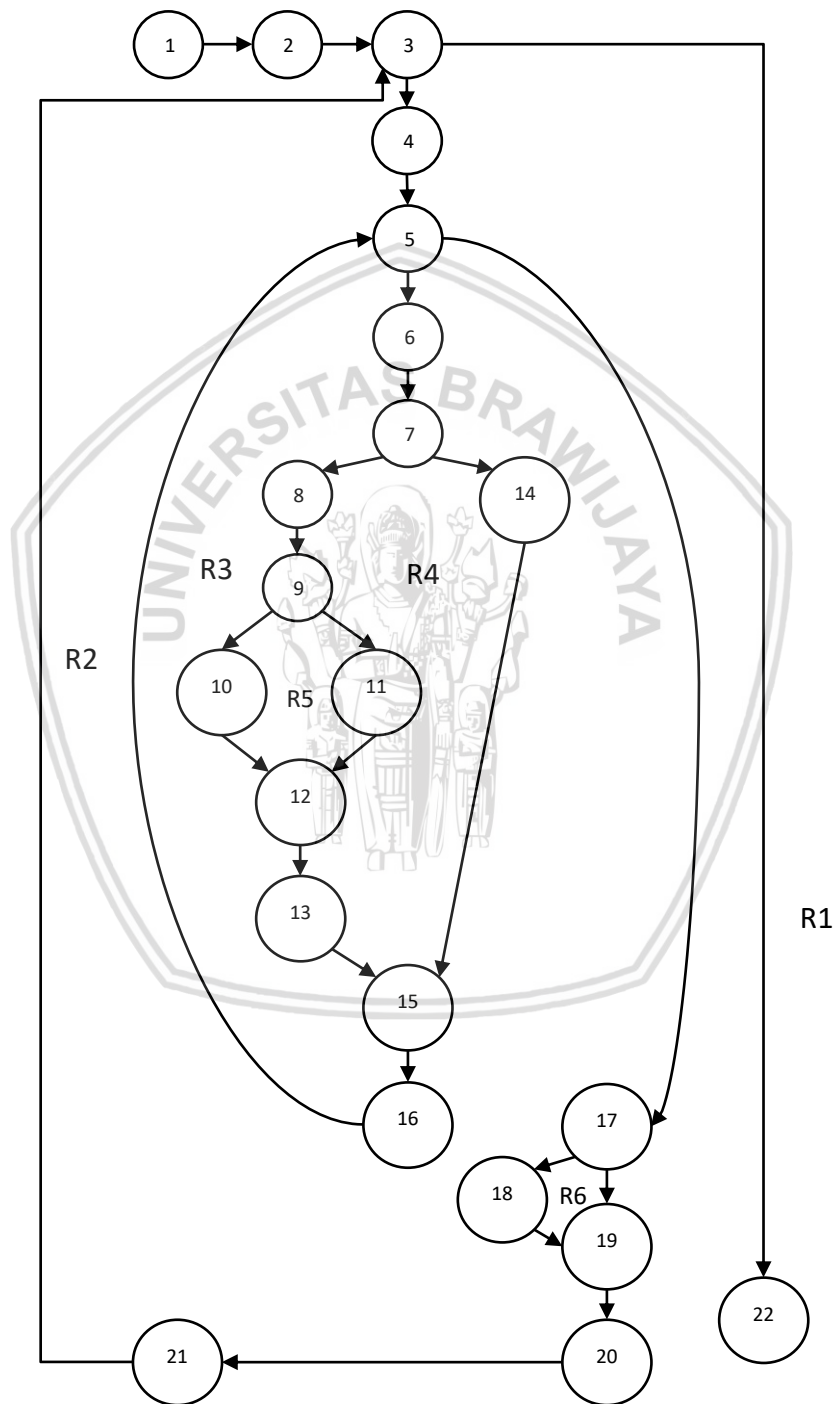
```

endif (19)
increment i (20)
end foreach (21)
end (22)

```

## 2. Basis Path Testing

### a. Flow Graph



Gambar 6.1 Flow Graph Method hitungStatistik

b. *Cyclomatic Complexity*

- a)  $V(G) = 6$ , ada 6 region
- b)  $V(G) = 26 \text{ edges} - 22 \text{ nodes} + 2 = 6$
- c)  $V(G) = 5 \text{ predicate nodes} + 1 = 6$

c. *Independent Path*

- a) Jalur 1 = 1 – 2 – 3 – 22
- b) Jalur 2 = 1 – 2 – 3 – 4 – 5 – 17 – 19 – 20 – 21 – 3 – 22
- c) Jalur 3 = 1 – 2 – 3 – 4 – 5 – 17 – 18 – 19 – 20 – 21 – 3 – 22
- d) Jalur 4 = 1 – 2 – 3 – 4 – 5 – 6 – 7 – 14 – 15 – 16 – 5 – 17 – 19 – 20 – 21 – 3 – 22
- e) Jalur 5 = 1 – 2 – 3 – 4 – 5 – 6 – 7 – 8 – 9 – 10 – 12 – 13 – 15 – 16 – 5 – 17 – 19 – 20 – 21 – 3 – 22
- f) Jalur 6 = 1 – 2 – 3 – 4 – 5 – 6 – 7 – 8 – 9 – 11 – 12 – 13 – 15 – 16 – 5 – 17 – 19 – 20 – 21 – 3 – 22

Tabel 6.2 Hasil Pengujian Unit *Method* hitungStatistik

No.	No. Jalur	Prosedur Uji	Expected Result	Result	Status
1.	1	Nilai variabel opd berisi 43 data, di mana opd = total jumlah opd	<i>Method</i> akan menyelesaikan perulangan	<i>Method</i> berhasil menyelesaikan perulangan	Valid
2.	2	Nilai variabel laporan berisi 2 data, di mana laporan = total jumlah laporan	<i>Method</i> akan menyelesaikan perulangan	<i>Method</i> berhasil menyelesaikan perulangan	Valid
3.	3	$j = 2$ , di mana $j \neq 0$	<i>Method</i> akan melakukan penghitungan nilai	<i>Method</i> berhasil melakukan penghitungan nilai	Valid
4.	4	tanggalMasuk = 0	<i>Method</i> akan memberikan nilai[i] = 0	<i>Method</i> berhasil memberikan nilai[i] = 0	Valid
5.	5	tanggalMasuk = 2, di mana tanggalMasuk $\neq 0$	<i>Method</i> akan memberikan nilai selisih = 10	<i>Method</i> berhasil	Valid

		selisih = 11, di mana selisih > 10		memberikan nilai selisih = 10	
6.	6	tanggalMasuk = 2, di mana tanggalMasuk != 0 selisih = 1, di mana selisih ≤ 10	Method akan memberikan nilai selisih = selisih	Method berhasil memberikan nilai selisih = selisih	Valid

### 6.1.2 Pengujian Unit *Method* hitungRekap

#### 1. Pseudocode

**Tabel 6.3 Pseudocode Method hitungRekap**

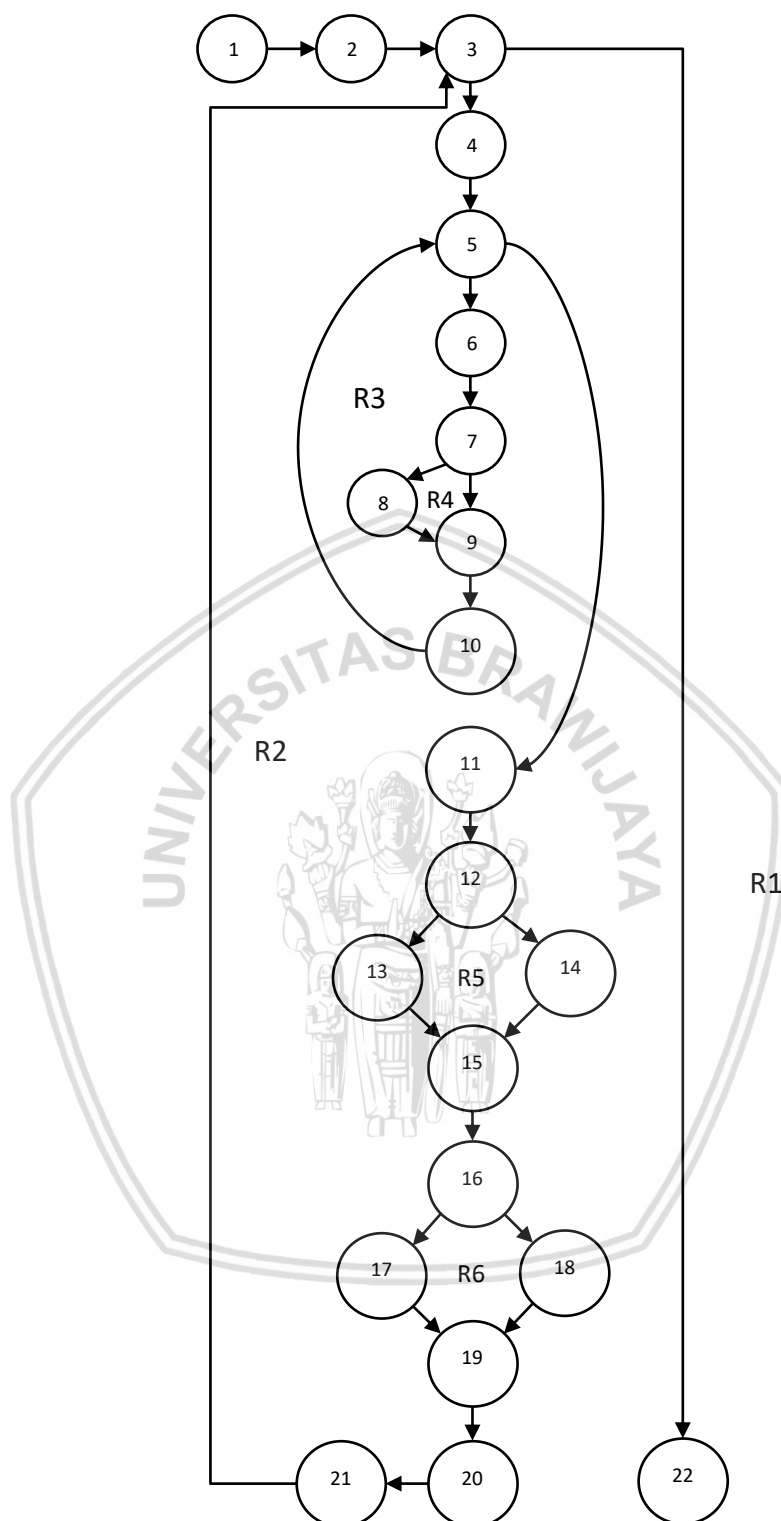
```

Start method hitungRekap (1)
waktu = memanggil method rekap
opd = data opd
i = 0
laporanArray = array()
lapDijawabArray = array()
lapTidakDijawabArray = array()
nilaiAkhir = array()
j = 0
foreach opd (3)
    laporan = ambil jumlah laporan berdasarkan bulan dan tahun dari variabel waktu
    laporanArray[i++] = laporan
    lap = ambil data laporan berdasarkan bulan dan tahun dari variabel waktu
    lapDijawabArray[j]=0
    foreach lap (5)
        lapDijawab = ambil laporan yang dijawab (6)
        if lapDijawab > 0 (7)
            lapDijawabArray[j] += 1 (8)
        endif (9)
    end foreach (10)
    lapTidakDijawabArray[j] = laporan-lapDijawabArray[j]
    nilai = lapDijawabArray[j]
    if nilai<0, (12) then 0, (13) else nilai (14)
    endif (15)
    if laporan != 0 (16)
        nilaiAkhir[j] = nilai/laporan*100 (17)
    else
        nilaiAkhir[j] = 0 } (18)
    end if (19)
    increment j (20)
end foreach (21)
end (22)

```

#### 2. Basis Path Testing

##### a. Flow Graph



**Gambar 6.2 Flow Graph Method hitungRekap**

*b. Cyclomatic Complexity*

a)  $V(G) = 6$ , ada 6 region

b)  $V(G) = 26 \text{ edges} - 22 \text{ nodes} + 2 = 6$

c)  $V(G) = 5 \text{ predicate nodes} + 1 = 6$

c. *Independent Path*

a) Jalur 1 = 1 – 2 – 3 – 22

b) Jalur 2 = 1 – 2 – 3 – 4 – 5 – 11 – 12 – 13 – 15 – 16 – 18 – 19 – 20 – 21 – 3 – 22

c) Jalur 3 = 1 – 2 – 3 – 4 – 5 – 11 – 12 – 13 – 15 – 16 – 17 – 19 – 20 – 21 – 3 – 22

d) Jalur 4 = 1 – 2 – 3 – 4 – 5 – 11 – 12 – 14 – 15 – 16 – 17 – 19 – 20 – 21 – 3 – 22

e) Jalur 5 = 1 – 2 – 3 – 4 – 5 – 11 – 12 – 14 – 15 – 16 – 18 – 19 – 20 – 21 – 3 – 22

f) Jalur 6 = 1 – 2 – 3 – 4 – 5 – 6 – 7 – 8 – 9 – 10 – 5 – 11 – 12 – 13 – 15 – 16 – 18 – 19 – 20 – 21 – 3 – 22

**Tabel 6.4 Hasil Pengujian Unit *Method* hitungRekap**

No.	No. Jalur	Prosedur Uji	<i>Expected Result</i>	<i>Result</i>	Status
1.	1	Nilai variabel opd berisi 43 data, di mana opd = total jumlah opd	<i>Method</i> akan menyelesaikan perulangan	<i>Method</i> berhasil menyelesaikan perulangan	Valid
2.	2	Nilai variabel lap berisi 2 data, di mana lap = total jumlah lap	<i>Method</i> akan menyelesaikan perulangan	<i>Method</i> berhasil menyelesaikan perulangan	Valid
3.	3	nilai = -1, di mana nilai < 0 laporan = 2, di mana laporan != 0	<i>Method</i> akan memberikan nilai = 0, lalu melakukan penghitungan nilaiAkhir	<i>Method</i> berhasil memberikan nilai = 0, lalu melakukan penghitungan nilaiAkhir	Valid
4.	4	nilai ≥ 0	<i>Method</i> akan memberikan nilai = nilai	<i>Method</i> berhasil memberikan nilai = nilai	Valid
5.	5	laporan = 0	<i>Method</i> akan memberikan nilaiAkhir = 0	<i>Method</i> berhasil memberikan nilaiAkhir = 0	Valid
6.	6	lapDijawab = 2, di mana lapDijawab > 0	<i>Method</i> akan melakukan	<i>Method</i> berhasil melakukan	Valid

			penghitungan lapDijawabArray	penghitungan lapDijawabArray	
--	--	--	---------------------------------	---------------------------------	--

6.1.3 Pengujian Unit *Method* teruskan

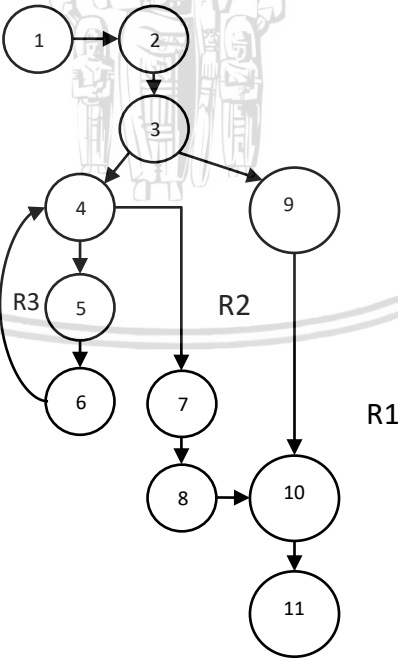
1. *Pseudocode*

Tabel 6.5 *Pseudocode Method* teruskan

Start method teruskan (1) opd = input(opd) (2) if opd not empty (3) foreach opd (4) data_insert = array(kode laporan, selected opd) } (5) masukkan data_insert ke database end foreach (6) data_update = array( "status_terusan" = "1" ) update data_update ke database Print alert 'Laporan Berhasil Diteruskan' (8) else Print alert 'Laporan Gagal Diteruskan' } (9) endif (10) end (11)
--

2. *Basis Path Testing*

a. *Flow Graph*



Gambar 6.3 *Flow Graph Method* teruskan

b. *Cyclomatic Complexity*

a)  $V(G) = 3$ , ada 3 region



- b)  $V(G) = 12 \text{ edges} - 11 \text{ nodes} + 2 = 3$   
 c)  $V(G) = 2 \text{ predicate nodes} + 1 = 3$   
 c. *Independent Path*  
 a) Jalur 1 = 1 – 2 – 3 – 9 – 10 – 11  
 b) Jalur 2 = 1 – 2 – 3 – 4 – 7 – 8 – 10 – 11  
 c) Jalur 3 = 1 – 2 – 3 – 4 – 5 – 6 – 4 – 7 – 8 – 10 – 11

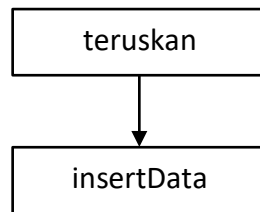
**Tabel 6.6 Hasil Pengujian Unit *Method* teruskan**

No.	No. Jalur	Prosedur Uji	<i>Expected Result</i>	<i>Result</i>	Status
1.	1	opd = 0, di mana opd = empty	<i>Method</i> akan menampilkan pesan gagal meneruskan laporan	<i>Method</i> berhasil menampilkan pesan gagal meneruskan laporan	Valid
2.	2	opd = 2, di mana opd = jumlah input opd	<i>Method</i> akan menyelesaikan perulangan	<i>Method</i> berhasil menyelesaikan perulangan	Valid
3.	3	opd = 2, di mana opd = !empty	<i>Method</i> akan menampilkan pesan berhasil meneruskan laporan	<i>Method</i> berhasil menampilkan pesan berhasil meneruskan laporan	Valid

## 6.2 Pengujian Integrasi

Pengujian integrasi dilakukan dengan menguji unit atau komponen yang berinteraksi dalam sistem untuk menghasilkan suatu fungsional. Pendekatan yang dilakukan pada pengujian integrasi ini adalah *top-down*.

Pada penelitian ini, pengujian integrasi dilakukan dengan menggunakan sampel fungsional meneruskan laporan. Dalam menjalankan fungsional ini, *Method* teruskan terintegrasi dengan *method* insertData sebagaimana dijelaskan pada Gambar 6.4.



**Gambar 6.4 Diagram Hierarki Pengujian Integrasi**

Pada *method* *teruskan*, dibuatkan sebuah *stub* yang berperan sebagai pengganti modul yang akan memanggil modul yang sedang diuji. Adapun langkah uji pengujian integrasi ini dijelaskan pada Tabel 6.7.

**Tabel 6.7 Langkah Uji Pengujian Integrasi**

No.	Langkah Uji	Keterangan
1	teruskan + insertData	<i>Method</i> <i>teruskan</i> dijalankan untuk menguji <i>method</i> <i>insertData</i> dengan menggunakan <i>stubInsertData</i> yang diberikan nilai input pada variabel <i>opd</i> berupa tipe data <i>String</i> berisi nilai “1” atau “0”.

**Tabel 6.8 Source Code Pengujian *Method* *stubInsertData* dengan Nilai *opd* “1”**

```

public function teruskan($kode_laporan){
    $data['kode_laporan'] = $kode_laporan;
    $opd = "1";
    if($opd=="1"){
        $res=$this->laporan_model->stubInsertData($opd);
        $this->session->set_flashdata("pesan",
            "<script>alert($res);</script>");
        redirect(base_url(). 'moderasi');
    } else{
        $this->session->set_flashdata("pesan",
            "<script>alert($opd);</script>");
        redirect(base_url(). 'moderasi');
    }
}
  
```

Tabel 6.8 merupakan *source code* dari pengujian integrasi *method* *teruskan* dengan *method* *stubInsertData*. Hasil dari penggunaan variabel *opd* dengan nilai “1” adalah *method* *teruskan* berhasil memanggil *stubInsertData* ketika nilai variabel *opd* “1” dengan nilai yang dikembalikan oleh *stubInsertData* sesuai dengan nilai variabel *opd* sebagaimana dapat dilihat pada Gambar 6.5.



**Gambar 6.5 Hasil Pengujian *Method* stubInsertData dengan Nilai opd “1”**

Lalu nilai variabel opd diubah menjadi “0” untuk menguji apakah *method* teruskan berhasil untuk tidak memanggil stubinsertData.

**Tabel 6.9 Source Code Pengujian *Method* stubInsertData dengan Nilai opd “0”**

```
public function teruskan($kode_laporan){
    $data['kode_laporan'] = $kode_laporan;
    $opd = "0";
    if($opd=="1"){
        $res=$this->laporan_model->stubInsertData($opd);
        $this->session->set_flashdata("pesan",
            "<script>alert($res);</script>");
        redirect(base_url(). 'moderasi');
    } else{
        $this->session->set_flashdata("pesan",
            "<script>alert($opd);</script>");
        redirect(base_url(). 'moderasi');
    }
}
```

Tabel 6.9 merupakan *source code* dari pengujian integrasi *method* teruskan dengan *method* stubInsertData. Hasil dari penggunaan variabel opd dengan nilai “0” adalah *method* teruskan berhasil masuk ke percabangan *else* dan tidak memanggil stubInsertData ketika nilai variabel opd adalah “0”. Nilai yang dikeluarkan sesuai dengan nilai opd sebagaimana dapat dilihat pada Gambar 6.6.



**Gambar 6.6 Hasil Pengujian *Method* stubInsertData dengan Nilai opd “0”**

Langkah selanjutnya yaitu mengintegrasikan *method* teruskan dengan *method* insertData untuk menguji apakah integrasi yang dihasilkan sama dengan pengujian yang telah dilakukan.

**Tabel 6.10 Source Code Method teruskan**

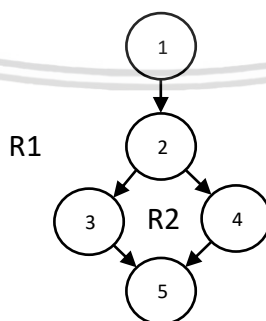
```

public function teruskan($kode_laporan){
    $data['kode_laporan'] = $kode_laporan;
    $opd = $this->input->post('opd');
    if(!empty($opd)){
        foreach($opd as $selected){
            $data_insert = array(
                "kode_laporan"    =>$kode_laporan,
                "id_user"         => $selected,
            );
            $this->laporan_model->InsertData($data_insert);
        }
        $data_update = array(
            "status_terusan" => "1"
        );
        $this->laporan_model->UpdateData($data_update, $data);
        $this->session->set_flashdata("pesan",
            "<script>alert('Laporan Berhasil Diteruskan');</script>");
        redirect(base_url().'moderasi');
    } else{
        $this->session->set_flashdata("pesan",
            "<script>alert('Laporan Gagal Diteruskan');</script>");
        redirect(base_url().'moderasi');
    }
}

```

Tabel 6.10 merupakan *method* teruskan yang secara keseluruhan diintegrasikan dengan *method* insertData. Berdasarkan proses pengujian integrasi *method* tersebut, dilakukan pengujian *white-box* dengan menggunakan *basis path testing*.

#### 1. Flow Graph

**Gambar 6.7 Flow Graph Integrasi Method insertData**

#### 2. Cyclomatic Complexity

- $V(G) = 2$ , ada 2 region
- $V(G) = 5 \text{ edges} - 5 \text{ nodes} + 2 = 2$
- $V(G) = 1 \text{ predicate nodes} + 1 = 2$

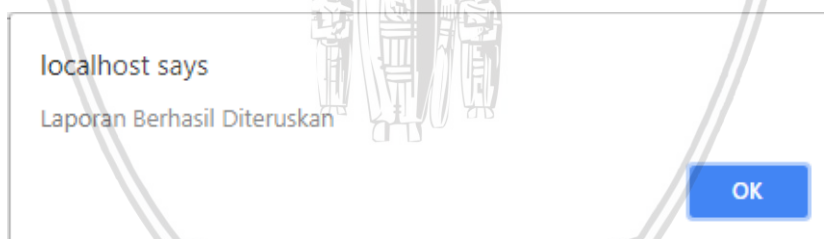
### 3. Independent Path

- a) Jalur 1 = 1 – 2 – 3 – 5
- b) Jalur 2 = 1 – 2 – 4 – 5

**Tabel 6.11 Hasil Pengujian Integrasi *Method* insertData**

No.	No. Jalur	Prosedur Uji	Expected Result	Result	Status
1.	1	opd = 1, di mana opd != empty	<i>Method</i> akan menampilkan pesan berhasil meneruskan laporan	<i>Method</i> akan menampilkan pesan berhasil meneruskan laporan	Valid
2.	2	opd = 0, di mana opd = empty	<i>Method</i> akan menampilkan pesan gagal meneruskan laporan	<i>Method</i> berhasil menampilkan pesan gagal meneruskan laporan	Valid

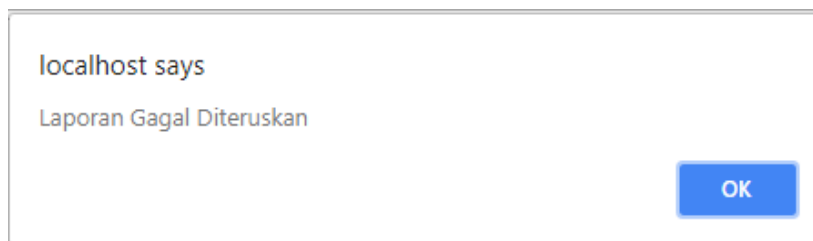
Langkah selanjutnya yaitu membandingkan hasil yang didapatkan pada saat menggunakan `stubInsertData` dengan hasil yang didapatkan pada saat menggunakan *method* insertData. Untuk mendapatkan nilai 1 seperti saat menggunakan *stub*, akan dilakukan simulasi dengan mengisi variabel `opd` dari input aktor dengan cara memilih satu OPD pada *checkbox*.



**Gambar 6.8 Hasil Pengujian Integrasi *Method* insertData dengan Nilai “1”**

Gambar 6.8 menunjukkan hasil pengujian integrasi antara *method* teruskan dan *method* insertData. Hasil dari input pada variabel `opd` adalah *method* teruskan berhasil memanggil *method* insertData ketika nilai variabel `opd` terisi yang dikembalikan oleh insertData dengan menampilkan pesan “Laporan Berhasil Diteruskan”.

Lalu untuk mendapatkan nilai 0 seperti saat menggunakan *stub*, akan dilakukan simulasi dengan mengisi variabel `opd` dari input aktor dengan nilai *null* dengan cara tidak memilih OPD pada *checkbox*.



**Gambar 6.9 Hasil Pengujian Integrasi *Method* insertData dengan Nilai “0”**

Gambar 6.9 menunjukkan hasil pengujian integrasi antara *method* teruskan dan *method* insertData. Hasil dari input pada variabel opd adalah *method* teruskan berhasil untuk tidak memanggil *method* insertData ketika nilai variabel opd tidak terisi dengan menampilkan pesan “Laporan Gagal Diteruskan”. Dari pengujian integrasi yang telah dilakukan, maka dapat disimpulkan bahwa *method* teruskan dan *method* insertData telah berhasil terintegrasi.

### 6.3 Pengujian Validasi

Pengujian validasi merupakan pengujian yang dilakukan untuk memastikan bahwa perangkat lunak yang dibangun sudah benar sesuai dengan kebutuhan yang telah didefinisikan sebelumnya. Pengujian ini dilakukan dengan metode *black-box* dengan jenis *scenario-based testing*.

#### 6.3.1 Pengujian Validasi Mendaftar

1. Kasus Uji Mendaftar

**Tabel 6.12 Kasus Uji Mendaftar**

<b>Nama Kasus Uji</b>	Kasus Uji Mendaftar
<b>Kode Kebutuhan</b>	WADUL_F01
<b>Prosedur</b>	<ol style="list-style-type: none"> <li>1. Mengakses halaman daftar.</li> <li>2. Memasukkan nama lengkap = Richa Amalia.</li> <li>3. Memasukkan NIK = 1451502001111471.</li> <li>4. Memasukkan nomor <i>handphone</i> = 085707957017</li> <li>5. Memasukkan <i>username</i> = richa.</li> <li>6. Memasukkan <i>password</i> = 1122.</li> <li>7. Menekan tombol daftar.</li> </ol>
<b>Hasil yang diharapkan</b>	Sistem akan menampilkan pesan berhasil daftar.
<b>Hasil</b>	Sistem menampilkan pesan berhasil daftar.
<b>Status</b>	Valid

2. Kasus Uji Mendaftar Alternatif 1

**Tabel 6.13 Kasus Uji Mendaftar Alternatif 1**

<b>Nama Kasus Uji</b>	Kasus Uji Mendaftar Alternatif 1
-----------------------	----------------------------------



<b>Kode Kebutuhan</b>	WADUL_F01
<b>Prosedur</b>	<ol style="list-style-type: none"> <li>1. Mengakses halaman daftar.</li> <li>2. Memasukkan nama lengkap = Amalia P.</li> <li>3. Tidak memasukkan NIK.</li> <li>4. Memasukkan nomor <i>handphone</i> = 085708577710.</li> <li>5. Memasukkan <i>username</i> = amalia.</li> <li>6. Memasukkan <i>password</i> = 123.</li> <li>7. Menekan tombol daftar.</li> </ol>
<b>Hasil yang diharapkan</b>	Sistem akan memberikan peringatan untuk mengisi <i>field</i> yang kosong.
<b>Hasil</b>	Sistem memberikan peringatan untuk mengisi <i>field</i> yang kosong.
<b>Status</b>	Valid

3. Kasus Uji Mendaftar Alternatif 2

**Tabel 6.14 Kasus Uji Mendaftar Alternatif 2**

<b>Nama Kasus Uji</b>	Kasus Uji Mendaftar Alternatif 2
<b>Kode Kebutuhan</b>	WADUL_F01
<b>Prosedur</b>	<ol style="list-style-type: none"> <li>1. Mengakses halaman daftar.</li> <li>2. Memasukkan nama lengkap = Amalia P.</li> <li>3. Memasukkan NIK = 1451502001111471.</li> <li>4. Memasukkan nomor <i>handphone</i> = 085708577710.</li> <li>5. Memasukkan <i>username</i> = amalia.</li> <li>6. Memasukkan <i>password</i> = 123.</li> <li>7. Menekan tombol daftar.</li> </ol>
<b>Hasil yang diharapkan</b>	Sistem akan memberikan peringatan bahwa NIK sudah terdaftar.
<b>Hasil</b>	Sistem memberikan peringatan bahwa NIK sudah terdaftar.
<b>Status</b>	Valid

4. Kasus Uji Mendaftar Alternatif 3

**Tabel 6.15 Kasus Uji Mendaftar Alternatif 3**

<b>Nama Kasus Uji</b>	Kasus Uji Mendaftar Alternatif 3
<b>Kode Kebutuhan</b>	WADUL_F01
<b>Prosedur</b>	<ol style="list-style-type: none"> <li>1. Mengakses halaman daftar.</li> <li>2. Memasukkan nama lengkap = Amalia P.</li> <li>3. Memasukkan NIK = 14515020011114ay.</li> </ol>

	4. Memasukkan nomor <i>handphone</i> = 085708577710. 5. Memasukkan <i>username</i> = amalia. 6. Memasukkan <i>password</i> = 123. 7. Menekan tombol daftar.
<b>Hasil yang diharapkan</b>	Sistem akan memberikan peringatan bahwa NIK hanya boleh diisi angka.
<b>Hasil</b>	Sistem memberikan peringatan bahwa NIK hanya boleh diisi angka.
<b>Status</b>	Valid

5. Kasus Uji Mendaftar Alternatif 4

**Tabel 6.16 Kasus Uji Mendaftar Alternatif 4**

<b>Nama Kasus Uji</b>	Kasus Uji Mendaftar Alternatif 4
<b>Kode Kebutuhan</b>	WADUL_F01
<b>Prosedur</b>	1. Mengakses halaman daftar. 2. Memasukkan nama lengkap = Amalia P. 3. Memasukkan NIK = 145150200111. 4. Memasukkan nomor <i>handphone</i> = 085708577710. 5. Memasukkan <i>username</i> = amalia. 6. Memasukkan <i>password</i> = 123. 7. Menekan tombol daftar.
<b>Hasil yang diharapkan</b>	Sistem akan memberikan peringatan bahwa NIK kurang dari 16 karakter.
<b>Hasil</b>	Sistem memberikan peringatan bahwa NIK kurang dari 16 karakter.
<b>Status</b>	Valid

6. Kasus Uji Mendaftar Alternatif 5

**Tabel 6.17 Kasus Uji Mendaftar Alternatif 5**

<b>Nama Kasus Uji</b>	Kasus Uji Mendaftar Alternatif 5
<b>Kode Kebutuhan</b>	WADUL_F01
<b>Prosedur</b>	1. Mengakses halaman daftar. 2. Memasukkan nama lengkap = Amalia P. 3. Memasukkan NIK = 145150200111142341. 4. Memasukkan nomor <i>handphone</i> = 085708577710. 5. Memasukkan <i>username</i> = amalia. 6. Memasukkan <i>password</i> = 123.

	7. Menekan tombol daftar.
<b>Hasil yang diharapkan</b>	Sistem akan memberikan peringatan bahwa NIK lebih dari 16 karakter.
<b>Hasil</b>	Sistem memberikan peringatan bahwa NIK lebih dari 16 karakter.
<b>Status</b>	Valid

7. Kasus Uji Mendaftar Alternatif 6

**Tabel 6.18 Kasus Uji Mendaftar Alternatif 6**

<b>Nama Kasus Uji</b>	Kasus Uji Mendaftar Alternatif 6
<b>Kode Kebutuhan</b>	WADUL_F01
<b>Prosedur</b>	<ol style="list-style-type: none"> <li>1. Mengakses halaman daftar.</li> <li>2. Memasukkan nama lengkap = Amalia P.</li> <li>3. Memasukkan NIK = 1451502001111423.</li> <li>4. Memasukkan nomor <i>handphone</i> = 085708577710.</li> <li>5. Memasukkan <i>username</i> = richa.</li> <li>6. Memasukkan <i>password</i> = 123.</li> <li>7. Menekan tombol daftar.</li> </ol>
<b>Hasil yang diharapkan</b>	Sistem akan memberikan peringatan bahwa <i>username</i> sudah digunakan.
<b>Hasil</b>	Sistem memberikan peringatan bahwa <i>username</i> sudah digunakan.
<b>Status</b>	Valid

8. Kasus Uji Mendaftar Alternatif 7

**Tabel 6.19 Kasus Uji Mendaftar Alternatif 7**

<b>Nama Kasus Uji</b>	Kasus Uji Mendaftar Alternatif 7
<b>Kode Kebutuhan</b>	WADUL_F01
<b>Prosedur</b>	<ol style="list-style-type: none"> <li>1. Mengakses halaman daftar.</li> <li>2. Memasukkan nama lengkap = Amalia P.</li> <li>3. Memasukkan NIK = 1451502001111423.</li> <li>4. Memasukkan nomor <i>handphone</i> = 0857085777heh.</li> <li>5. Memasukkan <i>username</i> = amalia.</li> <li>6. Memasukkan <i>password</i> = 123.</li> <li>7. Menekan tombol daftar.</li> </ol>
<b>Hasil yang diharapkan</b>	Sistem akan memberikan peringatan bahwa nomor <i>handphone</i> hanya boleh diisi angka.

<b>Hasil</b>	Sistem memberikan peringatan bahwa nomor <i>handphone</i> hanya boleh diisi angka.
<b>Status</b>	Valid

## 9. Kasus Uji Mendaftar Alternatif 8

Tabel 6.20 Kasus Uji Mendaftar Alternatif 8

<b>Nama Kasus Uji</b>	Kasus Uji Mendaftar Alternatif 8
<b>Kode Kebutuhan</b>	WADUL_F01
<b>Prosedur</b>	<ol style="list-style-type: none"> <li>1. Mengakses halaman daftar.</li> <li>2. Memasukkan nama lengkap = Amalia P.</li> <li>3. Memasukkan NIK = 1451502001111423.</li> <li>4. Memasukkan nomor <i>handphone</i> = 08570.</li> <li>5. Memasukkan <i>username</i> = amalia.</li> <li>6. Memasukkan <i>password</i> = 123.</li> <li>7. Menekan tombol daftar.</li> </ol>
<b>Hasil yang diharapkan</b>	Sistem akan memberikan peringatan bahwa nomor <i>handphone</i> kurang dari 10 karakter.
<b>Hasil</b>	Sistem memberikan peringatan bahwa nomor <i>handphone</i> kurang dari 10 karakter.
<b>Status</b>	Valid

## 10. Kasus Uji Mendaftar Alternatif 9

Tabel 6.21 Kasus Uji Mendaftar Alternatif 9

<b>Nama Kasus Uji</b>	Kasus Uji Mendaftar Alternatif 9
<b>Kode Kebutuhan</b>	WADUL_F01
<b>Prosedur</b>	<ol style="list-style-type: none"> <li>1. Mengakses halaman daftar.</li> <li>2. Memasukkan nama lengkap = Amalia P.</li> <li>3. Memasukkan NIK = 1451502001111423.</li> <li>4. Memasukkan nomor <i>handphone</i> = 085708577777771.</li> <li>5. Memasukkan <i>username</i> = amalia.</li> <li>6. Memasukkan <i>password</i> = 123.</li> <li>7. Menekan tombol daftar.</li> </ol>
<b>Hasil yang diharapkan</b>	Sistem akan memberikan peringatan bahwa nomor <i>handphone</i> lebih dari 12 karakter.
<b>Hasil</b>	Sistem memberikan peringatan bahwa nomor <i>handphone</i> lebih dari 12 karakter.
<b>Status</b>	Valid

### 6.3.2 Pengujian Validasi *Login*

#### 1. Kasus Uji *Login* Sebagai *Member*

**Tabel 6.22 Kasus Uji *Login* Sebagai *Member***

<b>Nama Kasus Uji</b>	Kasus Uji <i>Login</i> Sebagai <i>Member</i>
<b>Kode Kebutuhan</b>	WADUL_F02
<b>Prosedur</b>	1. Memasukkan <i>username</i> = richa. 2. Memasukkan <i>password</i> = 1122. 3. Menekan tombol <i>login</i> .
<b>Hasil yang diharapkan</b>	Sistem akan menampilkan halaman utama <i>member</i> .
<b>Hasil</b>	Sistem menampilkan halaman utama <i>member</i> .
<b>Status</b>	Valid

#### 2. Kasus Uji *Login* Sebagai Admin

**Tabel 6.23 Kasus Uji *Login* Sebagai Admin**

<b>Nama Kasus Uji</b>	Kasus Uji <i>Login</i> Sebagai Admin
<b>Kode Kebutuhan</b>	WADUL_F02
<b>Prosedur</b>	1. Memasukkan <i>username</i> = admin. 2. Memasukkan <i>password</i> = admin. 3. Menekan tombol <i>login</i> .
<b>Hasil yang diharapkan</b>	Sistem akan menampilkan halaman utama admin.
<b>Hasil</b>	Sistem menampilkan halaman utama admin.
<b>Status</b>	Valid

#### 3. Kasus Uji *Login* Sebagai OPD

**Tabel 6.24 Kasus Uji *Login* Sebagai OPD**

<b>Nama Kasus Uji</b>	Kasus Uji <i>Login</i> Sebagai OPD
<b>Kode Kebutuhan</b>	WADUL_F02
<b>Prosedur</b>	1. Memasukkan <i>username</i> = dukcapil. 2. Memasukkan <i>password</i> = dukcapil. 3. Menekan tombol <i>login</i> .
<b>Hasil yang diharapkan</b>	Sistem akan menampilkan halaman utama OPD.
<b>Hasil</b>	Sistem menampilkan halaman utama OPD.
<b>Status</b>	Valid

#### 4. Kasus Uji *Login* Alternatif 1

Tabel 6.25 Kasus Uji *Login* Alternatif 1

<b>Nama Kasus Uji</b>	Kasus Uji <i>Login</i> Alternatif 1
<b>Kode Kebutuhan</b>	WADUL_F02
<b>Prosedur</b>	1. Memasukkan <i>username</i> = richa. 2. Menekan tombol <i>login</i> .
<b>Hasil yang diharapkan</b>	Sistem akan memberikan peringatan untuk mengisi <i>field</i> yang kosong.
<b>Hasil</b>	Sistem memberikan peringatan untuk mengisi <i>field</i> yang kosong.
<b>Status</b>	Valid

5. Kasus Uji *Login* Alternatif 2Tabel 6.26 Kasus Uji *Login* Alternatif 2

<b>Nama Kasus Uji</b>	Kasus Uji <i>Login</i> Alternatif 2
<b>Kode Kebutuhan</b>	WADUL_F02
<b>Prosedur</b>	1. Memasukkan <i>username</i> = richie. 2. Memasukkan <i>password</i> = 1122. 3. Menekan tombol <i>login</i> .
<b>Hasil yang diharapkan</b>	Sistem akan menampilkan pesan "Username belum terdaftar!".
<b>Hasil</b>	Sistem menampilkan pesan "Username belum terdaftar!".
<b>Status</b>	Valid

6. Kasus Uji *Login* Alternatif 3Tabel 6.27 Kasus Uji *Login* Alternatif 3

<b>Nama Kasus Uji</b>	Kasus Uji <i>Login</i> Alternatif 3
<b>Kode Kebutuhan</b>	WADUL_F02
<b>Prosedur</b>	1. Memasukkan <i>username</i> = richa. 2. Memasukkan <i>password</i> = 1234. 3. Menekan tombol <i>login</i> .
<b>Hasil yang diharapkan</b>	Sistem akan menampilkan pesan "Password salah!".
<b>Hasil</b>	Sistem menampilkan pesan "Password salah!".
<b>Status</b>	Valid



### 6.3.3 Pengujian Validasi Mencari Laporan

#### 1. Kasus Uji Mencari Laporan

**Tabel 6.28 Kasus Uji Mencari Laporan**

<b>Nama Kasus Uji</b>	Kasus Uji Mencari Laporan
<b>Kode Kebutuhan</b>	WADUL_F03
<b>Prosedur</b>	<ol style="list-style-type: none"> <li>1. Mengakses halaman cari laporan.</li> <li>2. Memasukkan kata kunci = jalan.</li> <li>3. Memasukkan subjek laporan = jalan rusak.</li> <li>4. Memasukkan nama pengirim = richa.</li> <li>5. Menekan tombol cari.</li> </ol>
<b>Hasil yang diharapkan</b>	Sistem akan menampilkan hasil pencarian.
<b>Hasil</b>	Sistem menampilkan hasil pencarian.
<b>Status</b>	Valid

#### 2. Kasus Uji Mencari Laporan Alternatif 1

**Tabel 6.29 Kasus Uji Mencari Laporan Alternatif 1**

<b>Nama Kasus Uji</b>	Kasus Uji Mencari Laporan Alternatif 1
<b>Kode Kebutuhan</b>	WADUL_F03
<b>Prosedur</b>	<ol style="list-style-type: none"> <li>1. Mengakses halaman cari laporan.</li> <li>2. Memasukkan subjek laporan = jalan rusak.</li> <li>3. Memasukkan nama pengirim = richa.</li> <li>4. Menekan tombol cari.</li> </ol>
<b>Hasil yang diharapkan</b>	Sistem akan memberikan peringatan untuk mengisi <i>field</i> yang kosong.
<b>Hasil</b>	Sistem memberikan peringatan untuk mengisi <i>field</i> yang kosong.
<b>Status</b>	Valid

### 6.3.4 Pengujian Validasi Melihat Status Tahapan Laporan

#### 1. Kasus Uji Melihat Status Tahapan Laporan

**Tabel 6.30 Kasus Uji Melihat Status Tahapan Laporan**

<b>Nama Kasus Uji</b>	Kasus Uji Melihat Status Tahapan Laporan
<b>Kode Kebutuhan</b>	WADUL_F04
<b>Prosedur</b>	<ol style="list-style-type: none"> <li>1. Mencari laporan.</li> <li>2. Memilih laporan.</li> </ol>

<b>Hasil yang diharapkan</b>	Sistem akan menampilkan rincian laporan dan status tahapan laporannya.
<b>Hasil</b>	Sistem menampilkan rincian laporan dan status tahapan laporannya.
<b>Status</b>	Valid

### 6.3.5 Pengujian Validasi Melihat Jawaban Laporan

1. Kasus Uji Melihat Jawaban Laporan

**Tabel 6.31 Kasus Uji Melihat Jawaban Laporan**

<b>Nama Kasus Uji</b>	Kasus Uji Melihat Jawaban Laporan
<b>Kode Kebutuhan</b>	WADUL_F05
<b>Prosedur</b>	1. Mencari laporan. 2. Memilih laporan.
<b>Hasil yang diharapkan</b>	Sistem akan menampilkan rincian laporan dan jawaban dari laporan tersebut.
<b>Hasil</b>	Sistem menampilkan rincian laporan dan jawaban dari laporan tersebut.
<b>Status</b>	Valid

### 6.3.6 Pengujian Validasi Melihat Statistik Kinerja OPD

1. Kasus Uji Melihat Statistik Kinerja OPD

**Tabel 6.32 Kasus Uji Melihat Statistik Kinerja OPD**

<b>Nama Kasus Uji</b>	Kasus Uji Melihat Statistik Kinerja OPD
<b>Kode Kebutuhan</b>	WADUL_F06
<b>Prosedur</b>	1. Mengakses halaman statistik.
<b>Hasil yang diharapkan</b>	Sistem akan menampilkan statistik kinerja OPD pada bulan dan tahun saat ini.
<b>Hasil</b>	Sistem menampilkan statistik kinerja OPD pada bulan dan tahun yang saat ini.
<b>Status</b>	Valid

### 6.3.7 Pengujian Validasi Mengubah Bulan dan Tahun Statistik

1. Kasus Uji Mengubah Bulan dan Tahun Statistik

**Tabel 6.33 Kasus Uji Mengubah Bulan dan Tahun Statistik**

<b>Nama Kasus Uji</b>	Kasus Uji Mengubah Bulan dan Tahun Statistik
-----------------------	--

<b>Kode Kebutuhan</b>	WADUL_F07
<b>Prosedur</b>	1. Mengakses halaman statistik. 2. Memilih bulan = Juni. 3. Memilih tahun = 2018. 4. Menekan tombol pilih.
<b>Hasil yang diharapkan</b>	Sistem akan menampilkan statistik kinerja OPD pada bulan dan tahun yang dipilih.
<b>Hasil</b>	Sistem menampilkan statistik kinerja OPD pada bulan dan tahun yang dipilih.
<b>Status</b>	Valid

2. Kasus Uji Mengubah Bulan dan Tahun Statistik Alternatif 1

**Tabel 6.34 Kasus Uji Mengubah Bulan dan Tahun Statistik Alternatif 1**

<b>Nama Kasus Uji</b>	Kasus Uji Mengubah Bulan dan Tahun Statistik Alternatif 1
<b>Kode Kebutuhan</b>	WADUL_F07
<b>Prosedur</b>	1. Mengakses halaman statistik. 2. Memilih bulan = Juni. 3. Menekan tombol pilih.
<b>Hasil yang diharapkan</b>	Sistem akan memberikan peringatan untuk mengisi <i>field</i> yang kosong.
<b>Hasil</b>	Sistem memberikan peringatan untuk mengisi <i>field</i> yang kosong.
<b>Status</b>	Valid

### 6.3.8 Pengujian Validasi Melihat Rekapitulasi Laporan

1. Kasus Uji Melihat Rekapitulasi Laporan

**Tabel 6.35 Kasus Uji Melihat Rekapitulasi Laporan**

<b>Nama Kasus Uji</b>	Kasus Uji Melihat Rekapitulasi Laporan
<b>Kode Kebutuhan</b>	WADUL_F08
<b>Prosedur</b>	1. Mengakses halaman rekapitulasi laporan.
<b>Hasil yang diharapkan</b>	Sistem akan menampilkan rekapitulasi laporan pada bulan dan tahun saat ini.
<b>Hasil</b>	Sistem menampilkan rekapitulasi laporan pada bulan dan tahun yang saat ini.
<b>Status</b>	Valid

### 6.3.9 Pengujian Validasi Mengubah Bulan dan Tahun Rekapitulasi

#### 1. Kasus Uji Mengubah Bulan dan Tahun Rekapitulasi

**Tabel 6.36 Kasus Uji Mengubah Bulan dan Tahun Rekapitulasi**

<b>Nama Kasus Uji</b>	Kasus Uji Mengubah Bulan dan Tahun Rekapitulasi
<b>Kode Kebutuhan</b>	WADUL_F09
<b>Prosedur</b>	1. Mengakses halaman rekapitulasi laporan. 2. Memilih bulan = Juni . 3. Memilih tahun = 2018. 4. Menekan tombol pilih.
<b>Hasil yang diharapkan</b>	Sistem akan menampilkan rekapitulasi laporan pada bulan dan tahun yang dipilih.
<b>Hasil</b>	Sistem menampilkan rekapitulasi laporan pada bulan dan tahun yang dipilih.
<b>Status</b>	Valid

#### 2. Kasus Uji Mengubah Bulan dan Tahun Rekapitulasi Alternatif 1

**Tabel 6.37 Kasus Uji Mengubah Bulan dan Tahun Rekapitulasi Alternatif 1**

<b>Nama Kasus Uji</b>	Kasus Uji Mengubah Bulan dan Tahun Rekapitulasi Alternatif 1
<b>Kode Kebutuhan</b>	WADUL_F09
<b>Prosedur</b>	1. Mengakses halaman rekapitulasi laporan. 2. Memilih bulan = Juni. 3. Menekan tombol pilih.
<b>Hasil yang diharapkan</b>	Sistem akan memberikan peringatan untuk mengisi <i>field</i> yang kosong.
<b>Hasil</b>	Sistem memberikan peringatan untuk mengisi <i>field</i> yang kosong.
<b>Status</b>	Valid

### 6.3.10 Pengujian Validasi Melaporkan Permasalahan

#### 1. Kasus Uji Melaporkan Permasalahan

**Tabel 6.38 Kasus Uji Melaporkan Permasalahan**

<b>Nama Kasus Uji</b>	Kasus Uji Melaporkan Permasalahan
<b>Kode Kebutuhan</b>	WADUL_F10
<b>Prosedur</b>	1. <i>Login</i> sebagai <i>member</i> . 2. Mengakses halaman formulir pelaporan.

	<ol style="list-style-type: none"> <li>Memasukkan subjek laporan.</li> <li>Memasukkan isi laporan.</li> <li>Memasukkan lokasi permasalahan.</li> <li>Mengunggah <i>file</i> (opsional).</li> <li>Menekan tombol kirim.</li> </ol>
<b>Hasil yang diharapkan</b>	Sistem akan menampilkan pesan “Data berhasil disimpan”.
<b>Hasil</b>	Sistem menampilkan pesan “Data berhasil disimpan”.
<b>Status</b>	Valid

2. Kasus Uji Melaporkan Permasalahan Alternatif 1

**Tabel 6.39 Kasus Uji Melaporkan Permasalahan Alternatif 1**

<b>Nama Kasus Uji</b>	Kasus Uji Melaporkan Permasalahan Alternatif 1
<b>Kode Kebutuhan</b>	WADUL_F10
<b>Prosedur</b>	<ol style="list-style-type: none"> <li><i>Login</i> sebagai <i>member</i>.</li> <li>Mengakses halaman formulir pelaporan.</li> <li>Memasukkan subjek laporan.</li> <li>Memasukkan lokasi permasalahan.</li> <li>Mengunggah <i>file</i> (opsional).</li> <li>Menekan tombol kirim.</li> </ol>
<b>Hasil yang diharapkan</b>	Sistem akan memberikan peringatan untuk mengisi <i>field</i> yang kosong.
<b>Hasil</b>	Sistem memberikan peringatan untuk mengisi <i>field</i> yang kosong.
<b>Status</b>	Valid

3. Kasus Uji Melaporkan Permasalahan Alternatif 2

**Tabel 6.40 Kasus Uji Melaporkan Permasalahan Alternatif 2**

<b>Nama Kasus Uji</b>	Kasus Uji Melaporkan Permasalahan Alternatif 2
<b>Kode Kebutuhan</b>	WADUL_F10
<b>Prosedur</b>	<ol style="list-style-type: none"> <li><i>Login</i> sebagai <i>member</i>.</li> <li>Mengakses halaman formulir pelaporan.</li> <li>Memasukkan subjek laporan.</li> <li>Memasukkan isi laporan.</li> <li>Memasukkan lokasi permasalahan.</li> <li>Mengunggah <i>file</i> berekstensi .pdf.</li> <li>Menekan tombol kirim.</li> </ol>

<b>Hasil yang diharapkan</b>	Sistem akan menampilkan pesan “Data gagal disimpan. File yang diupload harus berekstensi .jpg atau .png”.
<b>Hasil</b>	Sistem menampilkan pesan “Data gagal disimpan. File yang diupload harus berekstensi .jpg atau .png”.
<b>Status</b>	Valid

### 6.3.11 Pengujian Validasi Melihat Riwayat Laporan

1. Kasus Uji Melihat Riwayat Laporan

**Tabel 6.41 Kasus Uji Melihat Riwayat Laporan**

<b>Nama Kasus Uji</b>	Kasus Uji Melihat Riwayat Laporan
<b>Kode Kebutuhan</b>	WADUL_F11
<b>Prosedur</b>	1. <i>Login sebagai member.</i> 2. Mengakses halaman laporan saya.
<b>Hasil yang diharapkan</b>	Sistem akan menampilkan daftar laporan yang telah dikirimkan.
<b>Hasil</b>	Sistem menampilkan daftar laporan yang telah dikirimkan.
<b>Status</b>	Valid

### 6.3.12 Pengujian Validasi Memberi Respon Balik

1. Kasus Uji Memberi Respon Balik

**Tabel 6.42 Kasus Uji Memberi Respon Balik**

<b>Nama Kasus Uji</b>	Kasus Uji Memberi Respon Balik
<b>Kode Kebutuhan</b>	WADUL_F12
<b>Prosedur</b>	1. <i>Login sebagai member.</i> 2. Mengakses halaman laporan saya. 3. Memilih laporan. 4. Memasukkan isi respon. 5. Mengunggah <i>file</i> (opsional). 6. Menekan tombol kirim.
<b>Hasil yang diharapkan</b>	Sistem akan menampilkan respon yang telah dikirimkan.
<b>Hasil</b>	Sistem menampilkan respon yang telah dikirimkan.
<b>Status</b>	Valid



## 2. Kasus Uji Memberi Respon Balik Alternatif 1

Tabel 6.43 Kasus Uji Memberi Respon Balik Alternatif 1

<b>Nama Kasus Uji</b>	Kasus Uji Memberi Respon Balik Alternatif 1
<b>Kode Kebutuhan</b>	WADUL_F12
<b>Prosedur</b>	<ol style="list-style-type: none"> <li>1. <i>Login</i> sebagai <i>member</i>.</li> <li>2. Mengakses halaman laporan saya.</li> <li>3. Memilih laporan.</li> <li>4. Mengunggah <i>file</i> (opsional).</li> <li>5. Menekan tombol kirim.</li> </ol>
<b>Hasil yang diharapkan</b>	Sistem akan memberikan peringatan untuk mengisi <i>field</i> yang kosong.
<b>Hasil</b>	Sistem memberikan peringatan untuk mengisi <i>field</i> yang kosong.
<b>Status</b>	Valid

## 3. Kasus Uji Memberi Respon Balik Alternatif 2

Tabel 6.44 Kasus Uji Memberi Respon Balik Alternatif 2

<b>Nama Kasus Uji</b>	Kasus Uji Memberi Respon Balik Alternatif 2
<b>Kode Kebutuhan</b>	WADUL_F12
<b>Prosedur</b>	<ol style="list-style-type: none"> <li>1. <i>Login</i> sebagai <i>member</i>.</li> <li>2. Mengakses halaman laporan saya.</li> <li>3. Memilih laporan.</li> <li>4. Memasukkan isi respon</li> <li>5. Mengunggah <i>file</i> berekstensi .pdf.</li> <li>6. Menekan tombol kirim.</li> </ol>
<b>Hasil yang diharapkan</b>	Sistem akan menampilkan pesan “Respon gagal dikirim. File yang diupload harus berekstensi .jpg atau .png”.
<b>Hasil</b>	Sistem menampilkan pesan “Respon gagal dikirim. File yang diupload harus berekstensi .jpg atau .png”.
<b>Status</b>	Valid

## 6.3.13 Pengujian Validasi Menutup Laporan

## 1. Kasus Uji Menutup Laporan

Tabel 6.45 Kasus Uji Menutup Laporan

<b>Nama Kasus Uji</b>	Kasus Uji Menutup Laporan
-----------------------	---------------------------

<b>Kode Kebutuhan</b>	WADUL_F13
<b>Prosedur</b>	<ol style="list-style-type: none"> <li>1. <i>Login sebagai member.</i></li> <li>2. Mengakses halaman laporan saya.</li> <li>3. Memilih laporan.</li> <li>4. Menekan tombol tutup laporan.</li> <li>5. Menekan tombol OK pada dialog konfirmasi penutupan.</li> </ol>
<b>Hasil yang diharapkan</b>	Sistem akan mengubah status laporan menjadi "Tidak Aktif".
<b>Hasil</b>	Sistem mengubah status laporan menjadi "Tidak Aktif".
<b>Status</b>	Valid

2. Kasus Uji Menutup Laporan Alternatif 1

**Tabel 6.46 Kasus Uji Menutup Laporan Alternatif 1**

<b>Nama Kasus Uji</b>	Kasus Uji Menutup Laporan Alternatif 1
<b>Kode Kebutuhan</b>	WADUL_F13
<b>Prosedur</b>	<ol style="list-style-type: none"> <li>1. <i>Login sebagai member.</i></li> <li>2. Mengakses halaman laporan saya.</li> <li>3. Memilih laporan.</li> <li>4. Menekan tombol tutup laporan.</li> <li>5. Menekan tombol Cancel pada dialog konfirmasi penutupan.</li> </ol>
<b>Hasil yang diharapkan</b>	Sistem akan membatalkan penutupan laporan.
<b>Hasil</b>	Sistem membatalkan penutupan laporan.
<b>Status</b>	Valid

### 6.3.14 Pengujian Validasi Menjawab Laporan

1. Kasus Uji Menjawab Laporan

**Tabel 6.47 Kasus Uji Menjawab Laporan**

<b>Nama Kasus Uji</b>	Kasus Uji Menjawab Laporan
<b>Kode Kebutuhan</b>	WADUL_F14
<b>Prosedur</b>	<ol style="list-style-type: none"> <li>1. <i>Login sebagai OPD.</i></li> <li>2. Mengakses halaman laporan masuk.</li> <li>3. Memilih laporan.</li> <li>4. Memasukkan isi respon.</li> <li>5. Memilih estimasi tanggal.</li> <li>6. Memilih estimasi waktu.</li> </ol>

	7. Mengunggah <i>file</i> (opsional). 8. Menekan tombol kirim.
<b>Hasil yang diharapkan</b>	Sistem akan menampilkan respon yang telah dikirimkan.
<b>Hasil</b>	Sistem menampilkan respon yang telah dikirimkan.
<b>Status</b>	Valid

2. Kasus Uji Menjawab Laporan Alternatif 1

**Tabel 6.48 Kasus Uji Menjawab Laporan Alternatif 1**

<b>Nama Kasus Uji</b>	Kasus Uji Menjawab Laporan Alternatif 1
<b>Kode Kebutuhan</b>	WADUL_F14
<b>Prosedur</b>	1. <i>Login</i> sebagai OPD. 2. Mengakses halaman laporan masuk. 3. Memilih laporan. 4. Memilih estimasi tanggal. 5. Memilih estimasi waktu. 6. Mengunggah <i>file</i> (opsional). 7. Menekan tombol kirim.
<b>Hasil yang diharapkan</b>	Sistem akan memberikan peringatan untuk mengisi <i>field</i> yang kosong.
<b>Hasil</b>	Sistem memberikan peringatan untuk mengisi <i>field</i> yang kosong.
<b>Status</b>	Valid

3. Kasus Uji Menjawab Laporan Alternatif 2

**Tabel 6.49 Kasus Uji Menjawab Laporan Alternatif 2**

<b>Nama Kasus Uji</b>	Kasus Uji Menjawab Laporan Alternatif 2
<b>Kode Kebutuhan</b>	WADUL_F14
<b>Prosedur</b>	1. <i>Login</i> sebagai OPD. 2. Mengakses halaman laporan masuk. 3. Memilih laporan. 4. Memasukkan isi respon. 5. Memilih estimasi tanggal. 6. Memilih estimasi waktu. 7. Mengunggah <i>file</i> berekstensi .pdf. 8. Menekan tombol kirim.

<b>Hasil yang diharapkan</b>	Sistem akan menampilkan pesan “Respon gagal dikirim. File yang diupload harus berekstensi .jpg atau .png”.
<b>Hasil</b>	Sistem menampilkan pesan “Respon gagal dikirim. File yang diupload harus berekstensi .jpg atau .png”.
<b>Status</b>	Valid

### 6.3.15 Pengujian Validasi Melihat OPD

#### 1. Kasus Uji Melihat OPD

**Tabel 6.50 Kasus Uji Melihat OPD**

<b>Nama Kasus Uji</b>	Kasus Uji Melihat OPD
<b>Kode Kebutuhan</b>	WADUL_F15
<b>Prosedur</b>	1. <i>Login</i> sebagai admin. 2. Mengakses halaman kelola daftar OPD.
<b>Hasil yang diharapkan</b>	Sistem akan menampilkan daftar OPD.
<b>Hasil</b>	Sistem menampilkan daftar OPD.
<b>Status</b>	Valid

### 6.3.16 Pengujian Validasi Menambah OPD

#### 1. Kasus Uji Menambah OPD

**Tabel 6.51 Kasus Uji Menambah OPD**

<b>Nama Kasus Uji</b>	Kasus Uji Menambah OPD
<b>Kode Kebutuhan</b>	WADUL_F16
<b>Prosedur</b>	1. <i>Login</i> sebagai admin. 2. Mengakses halaman kelola daftar OPD. 3. Menekan tombol tambah OPD. 4. Memasukkan <i>username</i> = siwalan. 5. Memasukkan <i>password</i> . 6. Memasukkan nama OPD. 7. Memasukkan NIK. 8. Memasukkan nomor <i>handphone</i> . 9. Menekan tombol simpan.
<b>Hasil yang diharapkan</b>	Sistem akan memberi pesan “Data berhasil ditambahkan” dan menampilkan halaman daftar OPD yang baru.

<b>Hasil</b>	Sistem memberi pesan “Data berhasil ditambahkan” dan menampilkan halaman daftar OPD yang baru.
<b>Status</b>	Valid

2. Kasus Uji Menambah OPD Alternatif 1

**Tabel 6.52 Kasus Uji Menambah OPD Alternatif 1**

<b>Nama Kasus Uji</b>	Kasus Uji Menambah OPD Alternatif 1
<b>Kode Kebutuhan</b>	WADUL_F16
<b>Prosedur</b>	<ol style="list-style-type: none"> <li>1. <i>Login</i> sebagai admin.</li> <li>2. Mengakses halaman kelola daftar OPD.</li> <li>3. Menekan tombol tambah OPD.</li> <li>4. Memasukkan <i>password</i>.</li> <li>5. Memasukkan nama OPD.</li> <li>6. Memasukkan NIK.</li> <li>7. Memasukkan nomor <i>handphone</i>.</li> <li>8. Menekan tombol simpan.</li> </ol>
<b>Hasil yang diharapkan</b>	Sistem akan memberikan peringatan untuk mengisi <i>field</i> yang kosong.
<b>Hasil</b>	Sistem memberikan peringatan untuk mengisi <i>field</i> yang kosong.
<b>Status</b>	Valid

### 6.3.17 Pengujian Validasi Menghapus OPD

1. Kasus Uji Menghapus OPD

**Tabel 6.53 Kasus Uji Menghapus OPD**

<b>Nama Kasus Uji</b>	Kasus Uji Menghapus OPD
<b>Kode Kebutuhan</b>	WADUL_F17
<b>Prosedur</b>	<ol style="list-style-type: none"> <li>1. <i>Login</i> sebagai admin.</li> <li>2. Mengakses halaman kelola daftar OPD.</li> <li>3. Memilih OPD.</li> <li>4. Menekan tombol hapus OPD.</li> <li>5. Menekan tombol OK pada dialog konfirmasi penghapusan.</li> </ol>
<b>Hasil yang diharapkan</b>	Sistem akan memberi pesan “Data berhasil dihapus” dan menampilkan halaman daftar OPD yang baru.

<b>Hasil</b>	Sistem memberi pesan “Data berhasil dihapus” dan menampilkan halaman daftar OPD yang baru.
<b>Status</b>	Valid

2. Kasus Uji Menghapus OPD Alternatif 1

**Tabel 6.54 Kasus Uji Menghapus OPD Alternatif 1**

<b>Nama Kasus Uji</b>	Kasus Uji Menghapus OPD Alternatif 1
<b>Kode Kebutuhan</b>	WADUL_F17
<b>Prosedur</b>	<ol style="list-style-type: none"> <li>1. <i>Login</i> sebagai admin.</li> <li>2. Mengakses halaman kelola daftar OPD.</li> <li>3. Memilih OPD.</li> <li>4. Menekan tombol hapus OPD.</li> <li>5. Menekan tombol “Cancel” pada dialog konfirmasi penghapusan.</li> </ol>
<b>Hasil yang diharapkan</b>	Sistem akan membatalkan penghapusan OPD.
<b>Hasil</b>	Sistem membatalkan penghapusan OPD.
<b>Status</b>	Valid

**6.3.18 Pengujian Validasi Mengubah OPD**

1. Kasus Uji Mengubah OPD

**Tabel 6.55 Kasus Uji Mengubah OPD**

<b>Nama Kasus Uji</b>	Kasus Uji Mengubah OPD
<b>Kode Kebutuhan</b>	WADUL_F18
<b>Prosedur</b>	<ol style="list-style-type: none"> <li>1. <i>Login</i> sebagai admin.</li> <li>2. Mengakses halaman kelola daftar OPD.</li> <li>3. Memilih OPD.</li> <li>4. Menekan tombol edit OPD.</li> <li>5. Mengubah <i>username</i>.</li> <li>6. Mengubah <i>password</i>.</li> <li>7. Mengubah nama OPD.</li> <li>8. Mengubah NIK.</li> <li>9. Mengubah nomor <i>handphone</i>.</li> <li>10. Menekan tombol simpan.</li> </ol>
<b>Hasil yang diharapkan</b>	Sistem akan memberi pesan “Data berhasil diperbaharui” dan menampilkan halaman daftar OPD yang baru.



<b>Hasil</b>	Sistem memberi pesan “Data berhasil diperbaharui” dan menampilkan halaman daftar OPD yang baru.
<b>Status</b>	Valid

### 6.3.19 Pengujian Validasi Melakukan Moderasi Laporan

#### 1. Kasus Uji Melakukan Moderasi Laporan

**Tabel 6.56 Kasus Uji Melakukan Moderasi Laporan**

<b>Nama Kasus Uji</b>	Kasus Uji Melakukan Moderasi Laporan
<b>Kode Kebutuhan</b>	WADUL_F19
<b>Prosedur</b>	<ol style="list-style-type: none"> <li>1. <i>Login</i> sebagai admin.</li> <li>2. Mengakses halaman moderasi laporan.</li> <li>3. Memilih laporan.</li> <li>4. Menekan tombol <i>check</i>.</li> </ol>
<b>Hasil yang diharapkan</b>	Sistem akan mengubah status publikasi menjadi “Published”.
<b>Hasil</b>	Sistem mengubah status publikasi menjadi “Published”.
<b>Status</b>	Valid

#### 2. Kasus Uji Melakukan Moderasi Laporan Alternatif 1

**Tabel 6.57 Kasus Uji Melakukan Moderasi Laporan Alternatif 1**

<b>Nama Kasus Uji</b>	Kasus Uji Melakukan Moderasi Laporan Alternatif 1
<b>Kode Kebutuhan</b>	WADUL_F19
<b>Prosedur</b>	<ol style="list-style-type: none"> <li>1. <i>Login</i> sebagai admin.</li> <li>2. Mengakses halaman moderasi laporan.</li> <li>3. Memilih laporan.</li> <li>4. Menekan tombol silang.</li> </ol>
<b>Hasil yang diharapkan</b>	Sistem akan mengubah status publikasi menjadi “Unpublished”.
<b>Hasil</b>	Sistem mengubah status publikasi menjadi “Unpublished”.
<b>Status</b>	Valid

### 6.3.20 Pengujian Validasi Meneruskan Laporan

#### 1. Kasus Uji Meneruskan Laporan

**Tabel 6.58 Kasus Uji Meneruskan Laporan**

<b>Nama Kasus Uji</b>	Kasus Uji Meneruskan Laporan
<b>Kode Kebutuhan</b>	WADUL_F20
<b>Prosedur</b>	<ol style="list-style-type: none"> <li>1. <i>Login</i> sebagai admin.</li> <li>2. Mengakses halaman moderasi laporan.</li> <li>3. Memilih laporan.</li> <li>4. Menekan tombol teruskan.</li> <li>5. Memilih OPD = dukcapil.</li> <li>6. Menekan tombol teruskan laporan.</li> </ol>
<b>Hasil yang diharapkan</b>	Sistem akan menampilkan pesan “Laporan berhasil diteruskan”.
<b>Hasil</b>	Sistem menampilkan pesan “Laporan berhasil diteruskan”.
<b>Status</b>	Valid

## 2. Kasus Uji Meneruskan Laporan Alternatif 1

**Tabel 6.59 Kasus Uji Meneruskan Laporan Alternatif 1**

<b>Nama Kasus Uji</b>	Kasus Uji Meneruskan Laporan Alternatif 1
<b>Kode Kebutuhan</b>	WADUL_F20
<b>Prosedur</b>	<ol style="list-style-type: none"> <li>1. <i>Login</i> sebagai admin.</li> <li>2. Mengakses halaman moderasi laporan.</li> <li>3. Memilih laporan.</li> <li>4. Menekan tombol teruskan.</li> <li>5. Menekan tombol teruskan laporan.</li> </ol>
<b>Hasil yang diharapkan</b>	Sistem akan menampilkan pesan “Laporan gagal diteruskan”.
<b>Hasil</b>	Sistem menampilkan pesan “Laporan gagal diteruskan”.
<b>Status</b>	Valid

**6.3.21 Pengujian Validasi Logout**

## 1. Kasus Uji Logout

**Tabel 6.60 Kasus Uji Logout**

<b>Nama Kasus Uji</b>	Kasus Uji Logout
<b>Kode Kebutuhan</b>	WADUL_F21
<b>Prosedur</b>	<ol style="list-style-type: none"> <li>1. <i>Login</i> sebagai member/admin/OPD.</li> <li>2. Menekan tombol logout.</li> </ol>

<b>Hasil yang diharapkan</b>	Sistem akan menampilkan halaman utama.
<b>Hasil</b>	Sistem menampilkan halaman utama.
<b>Status</b>	Valid

## 6.4 Pengujian *Compatibility*

Pengujian *compatibility* dilakukan untuk mengetahui kemampuan sistem aplikasi pelaporan masyarakat dalam menampilkan halaman-halamannya sesuai dengan yang seharusnya pada beberapa *browser*, versi, dan *platform*. Pada penelitian ini, pengujian *compatibility* dilakukan menggunakan *tool* yang bernama SortSite.

The screenshot displays the SortSite Compatibility report for the URL <http://binamulabjn.com/wadul/>. The report is organized into tabs: Summary, Issues, Pages, Errors, Accessibility, Compatibility, Search, Standards, and Usability. The 'Compatibility' tab is active, showing a table of browser-specific issues.

Browser	Internet Explorer	Edge	Firefox	Safari	Opera	Chrome	iOS	Android
Version	8 9 10 11 16	60	≤ 10 11	51	66	≤ 9 10 11	≤ 3 4*	
Critical Issues	ⓘ ⓘ ⓘ ⓘ ⓘ	ⓘ	ⓘ	ⓘ ⓘ ⓘ ⓘ ⓘ	ⓘ	ⓘ	ⓘ ⓘ ⓘ ⓘ ⓘ	
Major Issues	ⓘ ⓘ ⓘ ⓘ ⓘ	ⓘ	ⓘ	ⓘ ⓘ ⓘ ⓘ ⓘ	ⓘ	ⓘ	ⓘ ⓘ ⓘ ⓘ ⓘ	
Minor Issues	ⓘ ⓘ ⓘ ⓘ ⓘ	ⓘ	ⓘ	ⓘ ⓘ ⓘ ⓘ ⓘ	ⓘ	ⓘ	ⓘ ⓘ ⓘ ⓘ ⓘ	

**Key:**  
 ⓘ Missing content or functionality  
 ⓘ Major layout or performance problems  
 ⓘ Minor layout or performance problems

\* Most Android devices from 4.1 onwards use Chrome as the default browser, older versions use the original Android browser

Priority	Description and URL	Guideline and Line#	Count
<b>Priority 1</b>	1 issues on 9 pages		
ⓘ ⓘ	Chrome and Firefox display multiple security warnings for password fields that don't use HTTPS	<a href="#">Chrome ≥ 56 Firefox ≥ 52</a>	9 pages
<b>Priority 3</b>	1 issues on 9 pages		
ⓘ ⓘ	The REQUIRED attribute is not supported in Internet Explorer 9 and earlier	<a href="#">Internet Explorer ≤ 9</a>	9 pages
▼	Expand all 2 issues		

**Gambar 6.10 Hasil Pengujian *Compatibility***

Gambar 6.1 merupakan hasil pengujian *compatibility* dengan menggunakan *tool* SortSite. Pengujian dilakukan terhadap delapan *browser* dengan total 15 versi dan 3 *platform*. Dalam pengujian ini, permasalahan dibagi menjadi tiga kategori, yaitu *critical issues*, *major issues*, dan *minor issues*. Hasil pengujian terhadap sistem aplikasi pelaporan masyarakat menunjukkan bahwa sistem berjalan dengan baik tanpa masalah pada 13 dari 15 versi *browser*. Terdapat dua *minor issues* pada *browser* Internet Explorer versi 8 dan 9, dan dua *critical issues* pada *browser* Firefox dan Chrome. *Critical issues* tersebut disebabkan karena tidak digunakannya HTTPS. Akan tetapi, apabila diakses langsung tanpa menggunakan HTTPS, sistem tetap dapat berjalan dengan baik tanpa ada fungsionalitas yang tidak berfungsi.

## BAB 7 PENUTUP

### 7.1 Kesimpulan

Dari hasil penelitian pengembangan sistem aplikasi pelaporan masyarakat berbasis web, maka dapat diperoleh kesimpulan sebagai berikut:

1. Berdasarkan hasil analisis kebutuhan, didapatkan 21 kebutuhan fungsional dan 1 kebutuhan non-fungsional yang dibutuhkan dalam sistem aplikasi pelaporan masyarakat berbasis web. Dalam tahap analisis kebutuhan, dilakukan analisis data yang dimodelkan dalam bentuk *Entity Relationship Diagram* (ERD) dan analisis kebutuhan yang dimodelkan dalam bentuk *use case diagram*. Pemodelan ERD yang dilakukan menghasilkan tiga entitas, yaitu *user*, *laporan*, dan *respon* yang masing-masing memiliki relasi antar entitas. Sedangkan pemodelan *use case diagram* menghasilkan 21 *use case* yang masing-masing dijelaskan dalam *use case scenario*.
2. Berdasarkan hasil perancangan, didapatkan perancangan data, *sequence diagram*, *class diagram*, perancangan algoritme, dan perancangan antarmuka. Perancangan data yang dibuat berupa *Physical Data Model* (PDM) yang menghasilkan empat tabel, yaitu *user*, *laporan*, *respon*, dan *terusan*. Tabel-tabel tersebut didapatkan dari hasil analisis data yang telah dibuat. *Sequence diagram* menjelaskan urutan proses dan interaksi antar objek dalam setiap fungsionalitas sistem. *Class diagram* menghasilkan perancangan kelas yang akan digunakan dalam pengembangan sistem berdasarkan arsitektur MVC (*Model-View-Controller*). Perancangan algoritme menghasilkan *pseudocode* yang akan digunakan dalam proses implementasi. Sedangkan perancangan antarmuka menghasilkan 17 *wireframe* yang menggambarkan antarmuka tiap halaman dalam sistem aplikasi pelaporan masyarakat.
3. Berdasarkan hasil implementasi, didapatkan spesifikasi pengembangan sistem yang menjelaskan spesifikasi perangkat keras dan perangkat lunak yang digunakan penulis dalam melakukan implementasi sistem, implementasi data berupa *Data Definition Language* (DDL), implementasi kode program yang didapatkan dari hasil perancangan algoritme, dan implementasi antarmuka yang didapatkan dari perancangan antarmuka.
4. Berdasarkan hasil pengujian, pengujian unit yang dilakukan dengan metode *white-box testing* menghasilkan nilai valid pada semua jalur uji. Pengujian integrasi yang dilakukan dengan metode *white-box testing* menghasilkan *method-method* yang diuji berhasil terintegrasi. Pengujian validasi yang dilakukan dengan metode *black-box testing* terhadap 21 kebutuhan fungsional menghasilkan nilai valid pada semua kasus uji. Sedangkan pengujian *compatibility* yang dilakukan menggunakan *tool* SortSite menunjukkan bahwa sistem dapat berjalan dengan baik pada 13 dari 15 versi *browser* yang diujikan.

## 7.2 Saran

Saran yang dapat diberikan untuk pengembangan sistem aplikasi pelaporan masyarakat selanjutnya antara lain:

1. Sistem dapat mengidentifikasi hari kerja dan hari libur.
2. Sistem menggunakan teknik *Information Retrieval* (IR) dalam mengembalikan hasil pencarian laporan.
3. Sistem dapat memberikan notifikasi bahwa laporan telah dijawab melalui SMS (*Short Message Service*) ke nomor *handphone* yang telah didaftarkan oleh *member*.



## DAFTAR PUSTAKA

- A. S., R. & Shalahuddin, M., 2016. *Rekayasa Perangkat Lunak Terstruktur dan Berorientasi Objek*. 4 penyunt. Bandung: Penerbit INFORMATIKA.
- Alve, S. et al., 2017. Web Application for Complaint Tracking and Resolving. *International Research Journal of Engineering and Technology (IRJET)*, 04(04), pp. 1396-1401.
- Arif, Eka & Hamintoko, 2017. *Penggalian Masalah Pelaporan Masyarakat di Kabupaten Pekalongan* [Wawancara] (25 Oktober 2017).
- BAPPENAS, 2010. *Laporan Kajian Manajemen Pengaduan Masyarakat dalam Pelayanan Publik*, Jakarta: Direktorat Aparatur Negara.
- Fowler, M., 2004. *UML Distilled Third Edition: A Brief Guide to the Standard Object Modeling Language*. 3 penyunt. Boston: Pearson Education, Inc.
- Hustinawati, Himawan, A. K. & Latifah, 2014. Performance Analysis Framework Codeigniter and CakePHP in Website Creation. *International Journal of Computer Applications*, 94(20), pp. 6-11.
- Idris, A., 2015. *Peranan Teknologi Informasi di Pemerintahan dan Pengenalan Industri Perangkat Lunak di Kabupaten Bireuen*, Aceh: Sekolah Tinggi Ilmu Ekonomi Kebangsaan Bireuen - Aceh.
- Informasi, K., 2013. *Komisi Informasi Pusat Republik Indonesia*. [Online] Available at: <https://komisiinformasi.go.id/news/view/keterbukaan-informasi-wujudkan-demokratisasi-pengelolaan-pemerintah> [Diakses 25 Oktober 2017].
- Kusuma, A. B. & Yulianto, L., 2014. Pembuatan Website Monitoring Keluhan Pelanggan Pada PT. Telkom Unit Layanan Ngadirojo. *Indonesian Journal on Networking and Security*, 3(4), pp. 2354-6654.
- Messmer, M., 2000. Careers. In: M. Messmer, ed. *Strategic Finance*. Montvale: Artville, p. 10.
- Nabil, D., Mosad, A. & Hefny, H. A., 2011. Web-Based Applications quality factors: A survey. *Egyptian Informatics Journal*, Issue 12, pp. 211-217.
- Nurmianto, E. & Siswanto, N., 2006. Perancangan Penilaian Kinerja Karyawan Berdasarkan Kompetensi Spencer dengan Metode Analytical Hierarchy Process (Studi Kasus di Sub Dinas Pengairan, Dinas Pekerjaan Umum, Kota Probolinggo). *Jurnal Teknik Industri*, 8(1), pp. 40-53.
- O'Docherty, M., 2005. *Object-Oriented Analysis and Design: Understanding System Development with UML 2.0*. 1 penyunt. West Sussex: John Wiley & Sons, Ltd.
- Oladimeji, P., 2007. *Levels of Testing*, Wales: University of Wales Swansea Computer Science Department.



Peranginangin, K., 2006. *Aplikasi Web dengan PHP dan MySQL*. 1 ed. Yogyakarta: Penerbit ANDI.

PowerMapper, 2015. *SortSite Technical Overview*. [Online] Available at: <https://www.powermapper.com/download/sortsite/sortsite-technical-overview.pdf> [Diakses 4 Juli 2018].

Pratama, A. N. W., 2010. *CodeIgniter: Cara Mudah Membangun Aplikasi PHP*. 1 penyunt. Jakarta Selatan: Mediakita.

Pressman, R. S., 2010. *Software Engineering: A Practitioners Approach*. 7th ed. New York: The McGraw-Hill Companies.

Sommerville, I., 2011. *Software Engineering*. 9th ed. Massachusetts: Pearson Education.

Wardana, 2010. *Menjadi Master PHP dengan Framework CodeIgniter*. 1 penyunt. Jakarta: Elex Media Komputindo.

Wellington, L. & Thomson, L., 2003. *PHP and MySQL Web Development*. 2 penyunt. Indianapolis: Sams Publishing.

